STRATA VISION 3d™

User Manual
Volume 2

FIRST PRINTING

December 1994

©1989-95 Strata Incorporated. All rights reserved. No part of this documentation may be reproduced, copied, transcribed, transmitted, stored in a retrieval system, or translated into any language in any form by any means without the written permission of Strata Incorporated.

ACKNOWLEDGEMENTS

Strata Incorporated acknowledges with gratitude the contributions of its development team who produced the Vision 3d product. Special thanks to Shelley Bracken, Product Manager, Joanne Austin, Quality Assurance, John Irwin, Val Clem and to an elite group of beta testers for their work.

- **Software written** by Gary Bringhurst**,** Jim Boyd, Brett Johnson, Brad Campbell, Bruce Gifford, Bruce Cardwell, John Adams, Brian Hill, Jim Wadley
- **Manual written** by Ron Woodland, Ken Perkins, Scott Summerill, Kathy Irvin
- **Interface design** by Ken Bringhurst, Kyle Maxwell, Lex de Azevedo
- **Package design** by Brett Marshall

STRATA

Strata Incorporated
2 W. St. George Blvd. • Ancestor Square, Suite 2100
St. George, Utah 84770 • (801) 628-5218

Technical support (voice): (801) 628-9751
Facsimile transmissions: (801) 628-9756

COPYRIGHT STATEMENT

These manuals, as well as the software described in them, are furnished under license and may only be used or copied in accordance with the terms of such license. The contents of these manuals are furnished for informational use only and is subject to change without notice. It should not be construed as a commitment by Strata Incorporated. Strata Incorporated assumes no responsibility or liability for any errors or inaccuracies that may appear in these manuals.

Except as permitted by license, no part of these manuals may be reproduced, stored in a retrieval system, or transmitted in any form or by any means — electronic, mechanical, or otherwise — without prior written permission of Strata Incorporated.

Please remember that the files provided for the various tutorials presented in the manuals may also be protected under copyright law. The unauthorized use of such artwork or images in your own projects could be a violation of the rights of the authors. Please be sure to obtain any permission required from such authors.

Strata, StrataVision 3d, and Vision 3d are trademarks of Strata Incorporated, which may be registered in some jurisdictions. Macintosh and Apple are registered trademarks of Apple Computer, Inc. Adobe, Adobe Photoshop, Adobe Illustrator, ATM, Adobe Type Manager, and PostScript are trademarks of Adobe System Incorporated. Canvas is a trademark of Deneba Systems, Inc. Other brand names and products mentioned in this manual are trademarks of their respective holders.

MacApp ©1985-1993 Apple Computer, Inc. Apple Computer makes no warranties whatsoever, either expressed or implied, regarding this product, including warranties with respect to its merchantability or its fitness for any particular purpose. The MacApp software is a trademark of Apple Computer, Inc. and is proprietary to Apple Computer, Inc., licensed to Strata Incorporated for distribution only for use in combination with StrataVision 3d.

For defense agencies, this product is subject to use, reproduction and disclosure restrictions as set forth in sub-paragraph (c)(1)(ii) of the Right in Technical Data and Computer Software clause at 252.227-7013.

For civilian agencies, this product is subject to use, reproduction and disclosure restrictions set forth in sub-paragraph (a) through (d) of the commercial Computer Software Restricted Rights clause at 52.227-19 and the limitations set forth in Strata Incorporated's standard commercial license agreement printed on the envelope containing the installation floppy diskettes for this software.

Unpublished rights reserved under copyright laws of the United States.

INTRODUCTION

Volume 2 of the User Manual will introduce you to the more complex features of StrataVision™ 3d. It will provide you with the information necessary to manipulate the shapes and models you learned to create in Volume 1. This volume will also provide you with the information required to add the finer details to your three-dimensional modeling.

Volume 2 includes chapters on Using Textures, Managing Lighting, Scripting Animations, and Rendering. It will be well worth the time spent to fully understand each of these features of Vision 3d. They will allow you to add dramatic impact to your final projects.

Vision 3d provides you with a wide variety of built-it textures, but it also allows you to create an endless variety of custom surface textures to use in your models. Mastering the use of textures will be well worth the effort.

Lighting is another important aspect of 3-D modeling. Vision 3d allows you to create customized lighting to meet your specific needs. It can add dramatic effects and remarkable realism to your models. The type and location of the lighting you use can completely change the appearance of a model, so spend some time experimenting with it to determine which effects best suit your needs.

The chapter on Scripting Animations gives you step by step instructions for creating your own animations. Keep in mind that rendering animations requires time...so please be patient. The results, however, will be well worth the wait.

When all your modeling is complete, it's time to render! Vision 3d offers you a wide variety of rendering methods to choose from. Play with each of the different renderers to compare their features, and then use the one that best fits your needs. The effects seen in your final rendered image will be determined by the renderer you choose.

The tutorials provided in this manual are presented to demonstrate how various functions of Vision 3d can be used. Using the tutorials will provide a better understanding of how to apply the many different techniques to your own 3-D modeling. Experiment with different features and discover the endless array of possibilities available to you. Remember, you are limited only by your imagination!

TABLE OF CONTENTS

CHAPTER 5: USING TEXTURES

Solid or surface textures 5.2
Solid textures require texture extensions 5.2
When a solid texture won't load 5.2
Advantages of surface textures 5.3
Making and editing textures 5.3
Surface textures — Explanation of fields 5.4
Example settings for popular textures 5.19
To review 5.19
Surface maps — explanation of fields 5.21
Using multiple surface maps 5.28
Making composite maps 5.32
Loading and removing Maps 5.44
Solid textures 5.45
Texture extensions explained 5.46
Handling textures in a model 5.50
Creating new textures 5.50
Loading textures 5.50
Applying textures 5.53
Application methods 5.53
Change Mapping dialog 5.53
Previewing the work as you go 5.55
How to make maps to correct scale and resolution 5.55
Quick Tutorials 5.64
Exercise 1: Label on a can 5.65
Exercise 2: Cracked eggshell 5.72
Textures on light sources 5.81
Giving lights color…and more 5.82
Using regular textures as a gel 5.84
Textures in environments 5.86
Textures Palette 5.87
Operation 5.89
Controls on the palette 5.90
Commands and dialogs used with Textures 5.92
Browse Textures command (Modeling menu) 5.92
Change Mapping dialog 5.94

Texture dialog 5.104
Show/Hide Textures Palette command (Windows menu) 5.114

CHAPTER 6: MANAGING LIGHTING

Properties of lights 6.2
Light icons are only placeholders 6.2
Global/Directional lights 6.4
Setting color 6.6
Setting intensity 6.7
Lighting palette 6.7
Toggling directional lights between hemispheres 6.9
Light meter 6.10
Point light sources 6.11
Setting color 6.12
Setting intensity 6.13
Setting attenuation 6.13
Spot light sources 6.15
Setting color 6.16
Setting intensity 6.16
Setting attenuation 6.18
Applying gels (all light sources) 6.19
Quick tutorials 6.21
Exercise 1: handling lights 6.21
Exercise 2: Using gels 6.27
Commands used in lighting control 6.39
Edit Primitive command (Modeling menu, ⌘B) 6.39
Lighting Effects command (Rendering menu, ⌘J) 6.39
Show/Hide Lighting Palette command (Windows menu) 6.41

CHAPTER 7: SCRIPTING ANIMATIONS

Hardware limitations 7.2
Concept of “actions” in creating motion 7.5
Actions that are time-dependent 7.5
Minimum requirements for animation 7.7
Event marker description 7.8
Visible motion paths 7.9
Using hierarchical scripting 7.10

An example: Scripting simple actions 7.11
Rendering the animation 7.30
The Animation palette 7.37
Selecting and moving event markers on the palette 7.39
Quick tutorials 7.40
Exercise 1: Simple motion 7.40
Exercise 2: A flying logo 7.44
QuickTime support 7.54
Using QuickTime for still images 7.56
Explanation of QT compressors 7.57
Animation extensions 7.59
Bounce 7.60
Convert to Path 7.63
Point At 7.70
Commands and controls used with animations 7.72
Animation windows (rendering and play back) 7.73
Cycle command (Animation menu) 7.75
New Event Marker command (Animation menu) 7.75
Playback Controls sub-menu (Animation menu) 7.75
Playback Frame Rate command (Animation menu) 7.80
Show Animation Paths command (Animation menu) 7.83

CHAPTER 8: RENDERING IMAGES

A digital “pinhole” camera 8.5
Better than a real pinhole camera 8.6
Greater focal range 8.6
Chasing after light rays 8.6
Perspective 8.8
Internal image handling 8.10
Suspending and restarting renderings 8.12
Setting and using preset rendering options 8.16
Saving rendering options by name 8.17
Deleting preset rendering options 8.19
Rendering options explained 8.19
Turning rendering effects on and off 8.19
Setting rendering parameters 8.20
About image resolution and anti-aliasing 8.22
Using the pixel averaging method 8.25

Rendering windows 8.27
Image Display Windows 8.28
Quick tutorial 8.29
Choosing the appropriate rendering method 8.35
How raytracing works 8.36
How RayPainting™ works 8.41
Comparison of the rendering algorithms 8.44
Charts of rendering effects supported (by rendering method) 8.51
Commands used with rendered images 8.54
Make Optimal Palette command (Edit menu) 8.62
Render command (Rendering menu, ⌘ Y) 8.65
Render Using command (Rendering menu) 8.66
Rendering Effects command (Rendering menu, ⌘ K) 8.68
Rendering Options command (Rendering menu) 8.71
Rendering Queue command (Rendering menu) 8.75
Suspend Rendering command (Rendering menu) 8.77

SUPPLEMENTAL MATERIAL

Glossary G.1
Appendix A A.1
Appendix B A.3
Appendix C A.4
Realistic Gems A.7

CHAPTER 5:
USING TEXTURES

Three-dimensional modeling deals not only with the geometry of objects, but also with the properties of their surfaces and how those surfaces interact with light and each other. The description and treatment of surfaces during rendering is at least as complex as modeling the objects themselves.

Because Vision 3d offers extensive surface-description capabilities, the most efficient approach to enhancing realism in rendered images is through textures. Which leads to the affirmation: ***"A good surface map is worth a thousand polygons."***

That means, whenever possible, you should rely on surface maps rather than creating explicit geometry to achieve detail. The resulting models are often more memory efficient and faster to render.

Textures are applied to objects to define their appearance when rendered. Several objects may use one texture, but only one texture can be applied to an object. The texture can be customized to each object's requirements with the "Change Mapping" dialog. When a texture is modified through its editing dialog boxes, all instances of that texture are automatically adjusted wherever it is used throughout the model.

Mastering the use of textures in Vision 3d, especially the advanced surface mapping features, is not a trivial matter. Properly executed, surface mapping enhances the realism of renderings such that they can become indistinguishable from an actual photograph. While this may seem a bold claim to make, nonetheless it is true. Many users vote their agreement again and again by choosing Vision 3d to render contest-winning images throughout the graphic arts industry.

If you invest the time necessary to learn Vision's finer points of texture handling, advanced mapping techniques, and multi-property composite maps, the results can be very rewarding. It will not come free, it will take time to learn and to do correctly. But nothing worth doing was ever easy. …If it was easy, everyone could do it, and you couldn't make any money at it!

This chapter contains the foundational and practical information upon which you can build your texture-creation skills. Once you begin applying the concepts, you will find that experimentation is really your best teacher, i.e. — you learn by doing. Trying out your own ideas just to “see what happens” can be a rewarding approach.

Solid or surface textures

Vision 3d supports both Solid (volumetric) and Surface (mapping) textures. While the steps to create either type of texture are similar, the way they work is quite different. And that difference is significant. Once you understand the difference, you will be better equipped to decide which type to use in any given situation.

Both types of textures have their own strengths and weaknesses. These will be identified in this chapter.

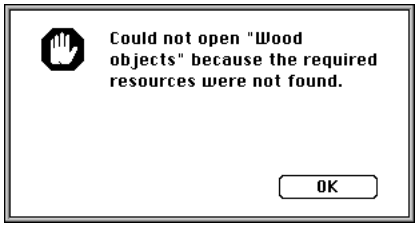
Solid textures require texture extensions

Solid textures offer greater technical precision and are computationally more efficient because they are generated by mathematical formulae. These are stored in the texture extension files within the Extensions folder of the StrataVision 3d application folder. You may create as many solid textures as you wish from one pattern stored in a texture extension. However, programming skills are usually required to create the formula stored in a texture extension.

Because solid textures are synthesized, they can be infinite in resolution and size. No matter how close you approach the surface with a solid texture, the pattern will not pixelize. No matter how big the object, a solid texture will cover its surface completely. On the other hand, solid textures can sometimes appear artificial, even cartoon-like, because they are mathematically derived. It takes a practiced eye to set up a solid texture’s parameters for a realistic feel.

When a solid texture won’t load

If you attempt to load a solid texture — either as part of an existing model or through the Load button in the textures browser — for which a texture extension cannot be found (because the extension is not in the Extensions folder), an alert similar to the following illustration appears and the operation is terminated.



Without a texture extension file, none of the textures that use that extension's formula can be loaded. If this occurs, something has changed within the StrataVision 3d application folder since the file was saved. You should check the list of texture extension stored within the Extensions folder and restore the complete set, if necessary. If the file is from another owner of Strata software, it is possible that a non-standard, third-party, or after-market texture extension was used.

Advantages of surface textures

In contrast, creating surface textures can be as simple as scanning a color picture, loading it into a texture-editing dialog box, and applying it to an object. They do not require the special support that solid textures do.

In more advanced functions, surface mapping offers the ability to control multiple surface properties at every point on the face of an object. This is done with specialized PICT images.

Properly used, surface textures can look quite real. Because the surface maps used are often digitized from actual objects, the features of surface textures are readily available to the average user and can be easily prepared with available software, such as paint programs with image processing tools.

Making and editing textures

Textures may be created in the active model at any time. Previously-built (and saved) textures may also be loaded from disk. Once a texture is defined in a model, it remains part of that model and is saved within the model file, even if it is not applied to anything. Textures may be deleted from a model only through the Browse Texture command. A delete button is provided in the Texture Browser dialog box to remove selected textures.

Once a texture is created or loaded into a model, it can be edited. It doesn't matter where the texture originated, or whether it is a surface or solid type texture…all textures are fully editable.

There are three ways available to open a texture for editing:

- **Using the Browse Textures dialog in the Textures menu.** In the textures browser, all loaded textures are listed by name. Buttons in this dialog box allow further manipulation, including editing.
- **Using the Edit button on the Textures palette.** In the palette, all textures are presented for selection by a representative preview images, whether they are loaded in the active model or not. Selecting a texture from the palette, then clicking on the Edit button automatically loads the textures and opens its editing dialog.
- **Double-clicking on a preview image on the Textures palette.** If a texture is loaded in the active model (no small floppy icon is visible in the upper right corner of the preview), double-clicking on its preview within the palette has the same effect as selecting the preview and clicking on the Edit button. If a texture is not loaded, double-clicking on its preview loads the texture into the active model.

When the editing of a texture is complete, i.e. — the OK button is clicked and the texture-editing dialog closes, the program returns to the location in the model that invoked the editing. This is especially significant in the case of the textures browser dialog box, which then allows you to conveniently select another texture for editing.

Surface textures — Explanation of fields

A texture is created by clicking on the New button of the Textures palette. By default, this opens the surface texture primary editing dialog. You have to take extra steps to create a solid texture.

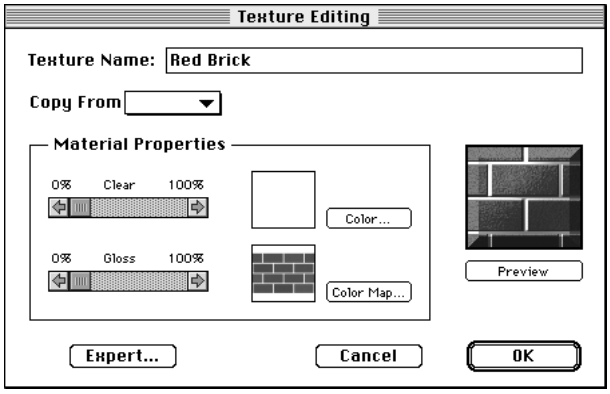
There are actually two dialogs available in the editing of surface textures: a primary and an expert. All surface textures use the same dialogs. The settings in these dialogs determine the properties of the texture named in the primary dialog.

PRIMARY LEVEL DIALOG

When you create a new texture or select an existing texture for editing, the primary "Texture Editing" dialog appears first. It

contains simple graphic controls and fields to set the various surface properties at an elementary level. For many textures, this primary dialog provides sufficient control.

The primary dialog is graphic in design, allowing you to interactively adjust the color, gloss and clear values and immediately see the effect of the changes in the dialog's preview area. You may modify the color, gloss and clear settings of any surface textures, even those containing surface maps. Adding color has the effect of tinting a color surface map.

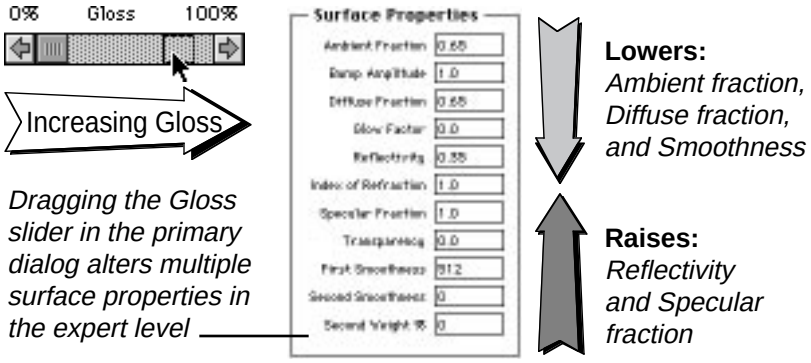


The fields presented in the primary surface texture dialog are as follows:

- **Texture Name.** Enter a unique name in this field. The name chosen is entirely your decision. Typically, the name you use would be descriptive of the texture defined, not the object to which you intend to apply it. This is especially recommended if more than one type of object will use the texture.
- **Copy From.** Use this pull-down menu to copy the properties from a previously defined texture that is also in the active model. Simply chose the desired texture from the menu. Releasing the mouse button copies all of that texture's settings to the open texture. The copied texture is unaffected, and any changes made apply only to the texture being edited.

- **Material Properties.** Two slider controls, titled Clear and Gloss allow you to control the properties of the texture in a simpler way.

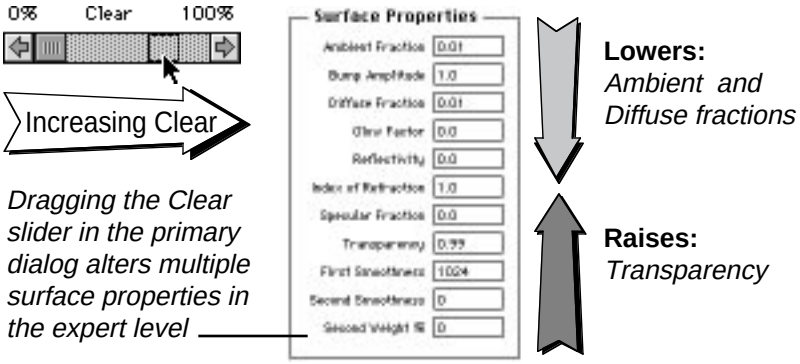
  The Gloss and Clear sliders actually control multiple surface properties. They are designed to adjust the appropriate surface properties in just the right combinations to produce the desired result. While these same properties are available in the expert level for individual adjustment, they require additional understanding to enter the appropriate combinations. Thus, the combination sliders are often preferable to direct property-by-property control.



Dragging the Gloss slider in the primary dialog alters multiple surface properties in the expert level

The Gloss slider affects the amount of ambient response, the amount of diffuse light reflected, the amount of specular light reflected, the amount of reflectivity, and the level of smoothness. Dragging the Gloss slider to the right increases the “shininess” of the surface; dragging it to the left reduces the shininess.

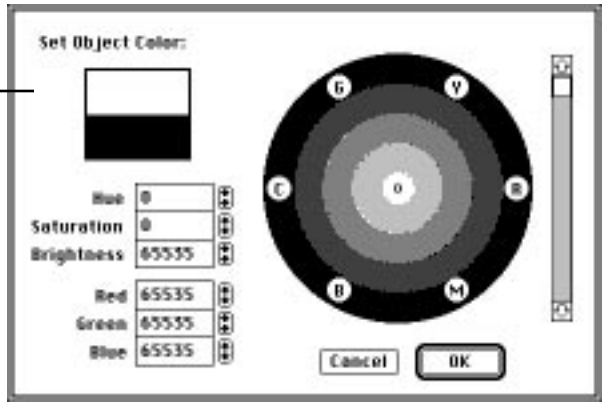
The Clear slider affects the amount of ambient response, the amount of diffuse light reflected, and the level of transparency. You can see that several properties are affected by both slider settings. Dragging the Clear slider to the right increases the transmission of light through the surface; dragging it to the left reduces the level of light transmission.



Dragging the Clear slider in the primary dialog alters multiple surface properties in the expert level

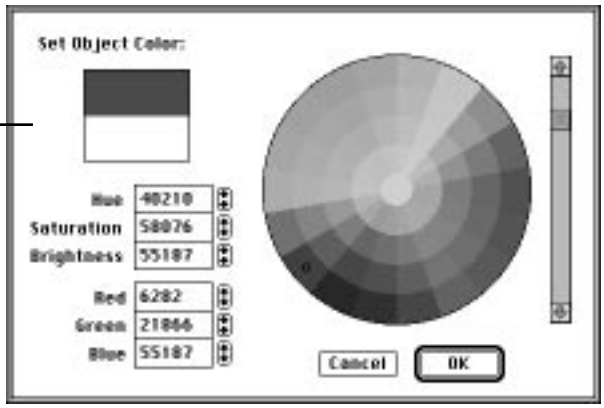
- **Color button.** Click on this button to choose a color as part of the texture definition. An Apple Color Picker dialog box opens. The configuration of this color picker dialog, and the number of colors that you may choose from depends on the current color depth of your monitor.

Apple Color Picker displayed on mono-chrome monitors



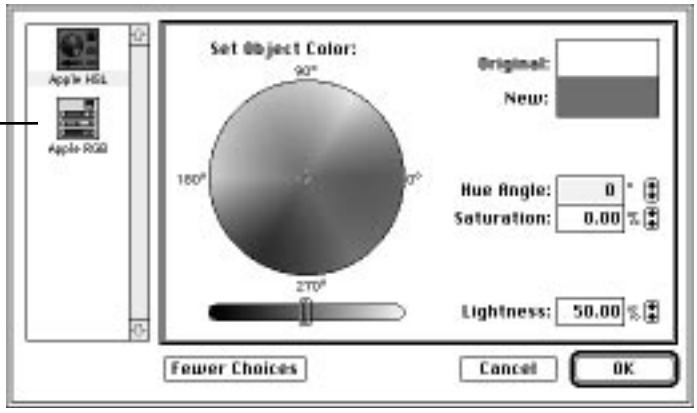
It also depends on the operating system that is running on your machine. Under System 7 or later, Apple Computer provides a separately-purchased extension called Color Sync™. This extension includes a new dialog with additional fields for choosing colors using various color spaces.

Apple Color Picker displayed on 8-bit (256 colors) monitors



Apple Color Picker displayed on 32-bit (16.7 colors) monitors

(This color picker is part of the Color Sync™ package. It is a System 7 extension that replaces the standard format.)

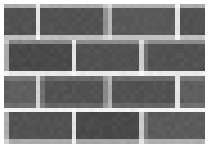


This field gives the texture a global color. When applied to an object, the entire surface of that object is tinted with this color. If used in conjunction with a color map, this color will also tint the colors present in the color map.

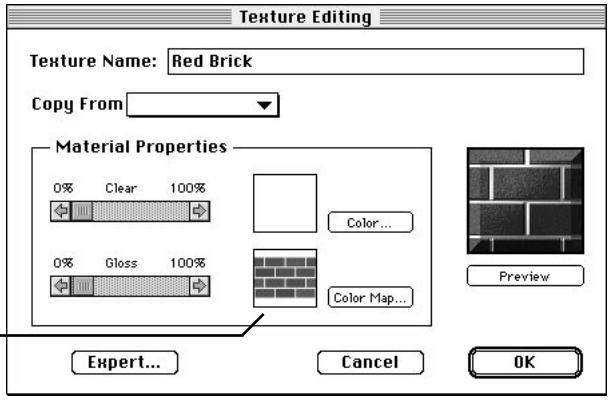
- **Color Map button.** Click on this button to bring a PICT image into the texture as a color map. When the texture is applied to an object, this map colors the surface pixel by pixel, based on the colors used in the map.

This is a small sample of a brick surface

(Note how the map's edges are set up to tile seamlessly, in all directions.)



Actual size: 2" high by 3" wide



It is loaded into the Color Map field of a surface texture's primary dialog

...then applied to a sphere using spherical mapping and normal tiling

(The map's coverage was set to 15% hor. and 30% vert., then tiled or repeated as needed to cover the entire surface.)

Actual size: 4" in diameter, 12.6" in circumference

- **Preview.** The preview area uses a generic object to show the texture's appearance. This preview object is separate from any objects in the active model.

- **Expert button.** Click on this button to open the second expert-level dialog. This is for adjusting the surface texture's properties in a more direct manner.

Once you're satisfied with the settings, click on the OK button or push the Return key to accept the texture you have defined and return to the model. If you make changes to an existing texture, then decide you want to revert to the previous settings, simply click on the "Cancel" button when exiting the dialog to discard any changes.

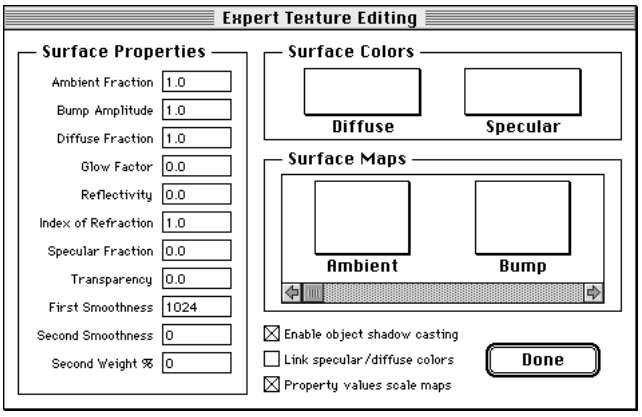
EXPERT LEVEL DIALOG

In the primary Texture Editing dialog, there is a button marked "Expert." Clicking on this button opens a secondary dialog providing specialized fields for defining more of the texture's parameters and using the advanced mapping features.

This dialog is divided into three main areas:

- **Surface Properties**
- **Surface Colors**
- **Surface Maps**

The specific combination of values and PICT images stored within a texture creates a unique surface appearance, which can then be mapped onto one or more objects or shapes.



Surface Properties fields are the principal settings that specify the overall appearance of a surface. Surface colors allow you to pick the global color properties for diffuse and specular reflections that may come from the surface. Surface maps work in conformity with

the surface properties and surface colors fields to provide pixel-by-pixel control of these properties over the surface of an object.

The expert-level dialog is where various types of surface maps are added to a texture. The various map fields use maps to provide pixel-by-pixel control on the surface of an object for their respective property.

For example: With a grayscale image in the Reflectivity map field, an object's surface becomes reflective only where the image is not black, and the reflectivity can vary with the levels of gray in the image. This is also the mechanism used in the other mapping fields.

SURFACE PROPERTIES DESCRIPTION

Vision's properties fields in the Expert dialog control parameters that directly affect how a surface responds or reflects light shining on the object to which it is applied. While experimentation is the most useful way to understand how these fields interact, the following field descriptions provide a beginning point and simple guidelines for their use:

- **Ambient Fraction.** This field determines how much of the available ambient light a specific surface reflects. Thus, even though the ambient light is constant throughout a given model, individual objects can react to that level differently. The range for this field is zero to one. A setting of zero means the surface will reflect no ambient light and will tend to be very dark (or black) in the shadows. A setting of one reflects all ambient light striking the surface.

Effects of the Ambient Fraction setting in texture definitions:

HIGH Ambient Fraction | *MEDIUM Ambient Fraction* | *LOW Ambient Fraction*

Notice that the lower the Ambient Fraction setting, the higher the contrast in the shaded areas of an object. The un-lit side is almost black in the image with a low ambient fraction.

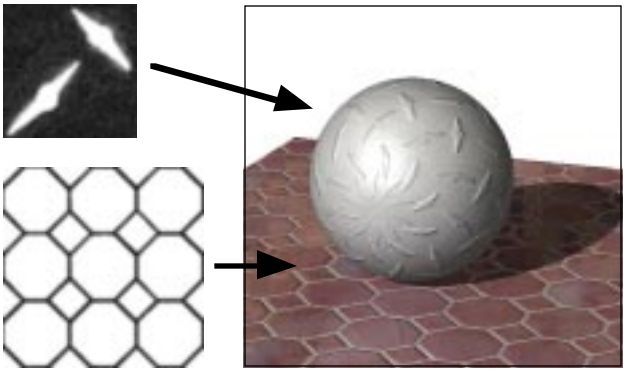
The level of ambient light in a model is set through the Lighting palette. Ambient light is a non-directional background illumination. It is part of the diffuse component of the lighting in a model.

- **Bump Amplitude.** This field works in conjunction with the Bump Map field. It sets the depth of surface variations specified by the bump map field. If no bump map is present in the texture, this setting has no effect. Bump mapping allows the generation of uneven surfaces too small or too detailed to be explicitly modeled.

  A bump map is created as a PICT image. It is a flat representation of a depressed or raised pattern you want applied to an object. The grayscale values in a bump map are used to vary the direction a surface's normal vectors point, thus changing the brightness of the reflected light from a surface. Bump mapping does not alter the actual geometry of an object, but rather creates the illusion of depressions or highs in the surface.

White areas in boilerplate's bump map creates raised areas on ball

Dark areas in ceramic tile's bump map creates depressions in floor for seams of the tiles



While there is no limit to this field's range, the optimum values usually fall between one and ten. The higher the value, the greater the surface variations…assuming a bump map is present, of course.

- **Diffuse Fraction.** Use this field to set the amount of diffuse light a surface reflects. A setting of zero reflects no diffuse light; a setting of one reflects all the diffuse light shining on it.

  Diffuse surfaces reflect light in all directions in a somewhat uniform manner, producing a matte or dull appearance. A surface's apparent brightness is set with this field. The range of

this field is zero to one. A zero value means no diffuse light is reflected and the surface appears dark, even when directly illuminated. Increasing the field value increases the amount of diffuse light reflected and makes the surface appear brighter.

NOTE: *Typically, diffuse and specular levels are mutually exclusive, i.e. — real world objects do not have high diffuse and high specular at the same time. These are the two main light reflection fields. (The Reflectivity field is a special case of Specular reflection). Other fields in the texture dialog may modify a surface, but can never completely obscure the basic nature of the light reflecting from it, as set by the Diffuse Fraction and Specular Fraction fields.*

- **Glow Factor.** Use this field to simulate the glow of a surface that is illuminated from behind, i.e. — light bulbs, translucent covers on lights, lamp shades, etc. Because the rendering algorithms cannot detect back lighting, this field allows you to create the effect. A setting of zero means no glow factor; there is no maximum value, but a value of one is usually enough to make the object fully glowing without any shading of the surface from outside light sources.

The light bulb in this table lamp is just a regular lathed object with a point light inside. It was given a texture with a high Glow Factor to simulate the light glowing within.

This field enhances the level of diffuse light that a surface reflects, even beyond what the full setting in the diffuse fraction

field would produce on its own. At higher levels, the object appears to glow or emit light. However, do not confuse a glow factor-enhanced surface with an actual light source. A glowing object will not illuminate surfaces adjacent to it. You may simulate the effect, however, by combining a glowing surface with appropriately placed light sources. This is often done to give the proper appearance of lighting fixtures and neon shapes.

- **Reflectivity.** Use this field to control the level of reflectivity on a surface. Reflectivity is a special phenomenon of specular light. At its shiniest — when the intensity of the reflected light is equal to that of the incident light and where the light being emitted has little diffuse component — the surface functions like a mirror.

  The color of the reflection is filtered by the Specular Color field if a color choice is present. If not, the reflection is in the color of the light striking the surface. The range of this field is zero to one.

  NOTE: *When creating a texture with high reflectivity, you must also reduce the ambient and diffuse fractions a corresponding amount. Otherwise, the ambient and diffuse light reflecting from a surface will obscure its reflectivity, no matter how high the reflectivity field is set. If there is no diffuse component at all, the surface becomes a perfect mirror without any haze from the diffuse light reflecting its surface. Such an object does not exist in the real world, so mirror-like textures usually also have a small amount of diffuse light present.*

- **Index of Refraction.** Use this field to control the amount of refraction visible through a transparent surface. The range of acceptable values starts at 1.0, which is air or no refraction. There is no upper limit, but in the real world there is no known transparent substance with a refraction index higher than 2.0. Glass is typically 1.5 to 1.6.

  The strength of the refraction depends on an object's geometry. As explained by the Laws of Physics, light refraction occurs at the surfaces of an object, where the light rays enter and exit the material. A transparent surface's refractive effect on the light coming from behind it depends on the angle of incidence and the shape of the surface at the points of incidence. The general effect of refraction is to shift the incident light to a parallel path as it exits the surface. At sea level, air has a refraction index of 1.00. Surfaces heavier than air have an index higher than one.

NOTE: In Vision 3d, REFRACTION TAKES PLACE ONLY WITH OBJECTS THAT ARE CONSTRUCTED WITH SINGLE-SIDED SURFACES. To understand the significance of this statement, please refer to the discussions in chapter 3 concerning the Extrude, Lathe, and Smooth modelers.

In the real world, the index of refraction for a substance is also a function of the wavelength of the incident light, such that different color components of the light will be refracted at different angles. In Vision 3d, only the raytracing algorithm calculates refraction. It does not consider wavelength in its refractive process. Instead, an average index of refraction is used and all colors are refracted the same amount.

In other words, *do not expect to be able to create a light spectrum on a wall by shining a white light through a prism-shaped object with a transparent and refractive texture applied. There are two reasons for this: First, as mentioned, the color of light sources is modeled as function of RGB color space, not wavelength; Second, for the same reason you cannot illuminate objects with back lighting, raytracing traces rays of light backwards from the view plane through the model. Thus, a light source "behind" a transparent object will not be detected and have no effect on the object until after that object has been rendered.*

- **Specular Fraction.** As with the Diffuse Fraction field counterpart, this control directly affects the level of specular light reflecting from a surface. A high specular component in the light normally gives strong highlights or pinpoints of light that give the surface a glossy appearance. The size and intensity of these "hot spots" also provide important visual cues to the relative smoothness of the surface. Specular highlights play a subtle, but important part in giving rendered images their photo-realistic "feel."

  Specular light reflections are typically in the color of the light source, unless a specular color is specified. The range of this field is zero to one.

- **Transparency.** Use this field to control the amount of transparency of a surface. The value range in this field is zero to one. Zero defines an opaque surface, a value of one is fully transparent or invisible. Light passing through a surface is called transmitted light. Like reflectivity, the more transparent a surface, the lower the ambient and diffuse fields should be. This is to remove light reflecting off an object's surface, which creates a haze that can obscure the transparency.

  When a transparent texture is applied, light coming from an object is modified to include contributions from light transmitted through it. In most cases, the light coming through an object is light-reflecting from surfaces of other objects. Diffuse and specular light passing through the surface are transmitted without adjustment, but are filtered by the Color field if a color choice is present. However, light coming from the surface is controlled by the Ambient and Diffuse fields.

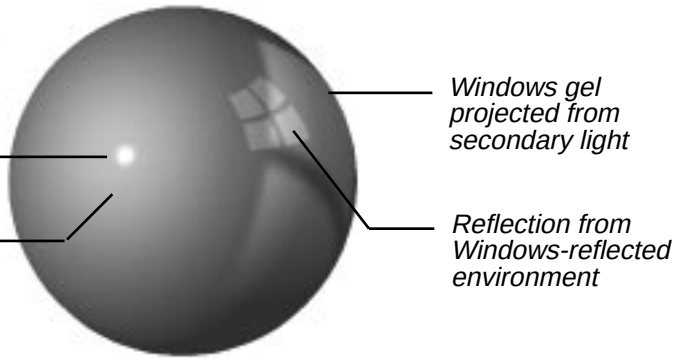
  NOTE: *When the transparency value is set to 1.0, a special case exists. All other surface properties are ignored, including color, ambient fraction, diffuse fraction, reflectivity, and specular fraction. In essence the surface (and the object to which it is applied) becomes invisible when rendered. If the transparency value is set to .99 or lower, then the values in these other fields are taken into account and will affect the surface's appearance.*

- **Smoothness and Weight Percentage.** The smoothness of a surface affects the nature of the light reflecting from it. The range is from zero to infinity. Higher values (500 and higher) mean smoother surfaces, which render with tighter specular highlights and sharper reflectivity. Lower values spread the specular hot

spots and slightly blur the reflected objects. Obviously, the higher the value, the shinier the surface appears.

The First Smoothness field is used by all textures. However, the second smoothness field is provided specifically for the creation of metallic surfaces. As you examine metallic objects around you, note how they have a high degree of specularity and that the hot spots (specular highlights) are in the color of the metal. The hot spots also tend to have halos around them. A metallic surface is established with high specular fraction (and correspondingly low ambient and diffuse fractions). The halo effect is simulated by using the first and second smoothness fields in combination, mixed by the Weight % field.

How to make metal objects look like metal

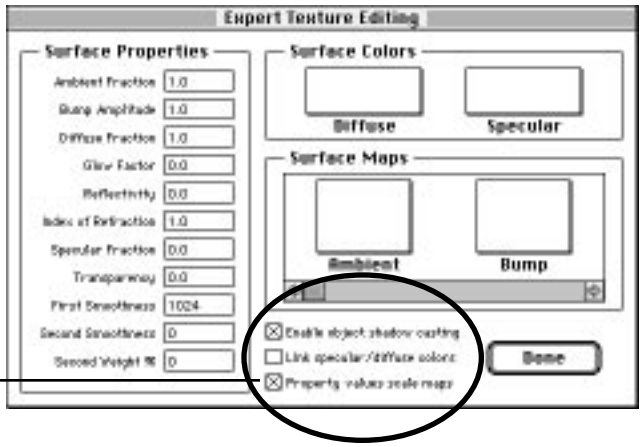

Specular highlight from primary light source
"Halo" around highlight
Halo is from a very low value (6) in the second smoothness field
Windows gel projected from secondary light
Reflection from Windows-reflected environment

Think of the First Smoothness field as the fineness of the machined surface and the Second Smoothness field as the coarseness of the crystalline structure of the material itself. The amount that the underlying crystal structure influences the surface is determined by the Weight % field. For metals, the first smoothness is typically set high (around 1,000), and the second smoothness is set very low (around 5), with a weight percentage usually between 25% and 50%. The settings in these smoothness fields have a subtle, yet observable influence on the light

reflected from a surface. The Weight % field is a percentage with a range from zero to 100%.

CHECKBOXES IN EXPERT DIALOG

Finally, there are three check boxes found at the bottom of the expert surface texture dialog box. They provide important control for special situations.



These check boxes change the basic behavior of a texture

An explanation of these check boxes follows:

- **Enable Object Shadow Casting.** This checkbox controls whether the object to which this texture is applied will cast a shadow when rendered with Raytracing or RayPainting. When turned off, no shadow will be cast, but it is also important to realize that light is allowed to pass through the object. This feature can be used for making light bulbs or covers for lighting fixtures (where the light source is inside). The default for this check box is on.
- **Link Specular/Diffuse Colors.** This check box controls whether the diffuse and specular colors are the same. It is provided primarily for use with metallic textures. As discussed earlier in this chapter, metallic objects reflect light in the color of the metal. When turned on, this feature enhances the metallic appearance of objects. The default setting is off.
- **Property Values Scale Maps.** No, this has nothing to do with real estate prices. This check box controls whether the surface

properties (on the left side of the expert dialog box) will be used to establish ranges or ceiling values (again nothing to do with real estate) within which the surface maps (on the right side of the dialog) will be scaled. When turned off, the surface maps have full range and the surface properties are not considered in the rendered image. The default setting for this check box is on.

Example settings for popular textures

A table is included in the appendix of this manual. It provides example values for the expert Surface Properties fields. Note the interaction between the various fields to achieve each type of surface, especially how the Ambient Fraction and Diffuse Fraction vary in relationship to Specular Fraction, Reflectivity, and Transparency.

Even if the particular texture you want is not represented there, you should be able to extrapolate the texture you want from these examples, but only if you understand the concepts involved.

To review

From this discussion, there are some general rules that indicate patterns in the settings of texture surface properties:

- **Surfaces become visible** when light is reflected off them toward the camera's view / rendering plane. The amount and type of light reflected tells you a lot about what material that object is made of.
- **Light reflecting from the surface of an object contains two components:** 1) diffuse and 2) specular. Surfaces may reflect just one type of light or both types in combination. A surface with a high diffuse component reflects light in all directions, causing it to appear matte or dull. Matte surfaces have the same brightness from all viewing angles. Surfaces with a strong specular component tend to reflect light only in the direction of the angle of reflection, giving them a shiny or gloss appearance. Generally, specular reflections appear not in the color of the surface, but rather the color of the light source. The exception to this is metallic objects.
- **Dull or matte textures** must have high ambient and diffuse fractions, with low specularity and reflectivity.

- **Shiny or reflective textures** have a high specularity fraction (and/or high reflectivity), with low ambient and diffuse settings.
- **Glowing objects** should have full ambient, full diffuse, no specular, and no reflectivity. Set the glow factor to .5 or higher, depending on how much you want it to glow. Glowing objects are not light sources, but can hold light sources inside or behind them. If you want a light source to illuminate other objects, you must turn off the first check box in the expert level to allow the light to pass through the surface of the object containing the light source.
- **For black objects,** a surface with no ambient, no diffuse, no specular, no reflectivity (and no glow factor) is essentially a "black hole." Since no light of any type reflects from its surface, it appears black. This is often a better technique than using the color black. If the object is too black — no details are visible on the surface — increase the ambient and diffuse slightly for a dark gray; more for a lighter gray.
- **For shiny black,** just increase the specular (and/or reflective) fields, but keep the ambient and diffuse at zero.
- **For a perfectly invisible surface,** use no ambient, no diffuse, no specular, and no reflectivity, but set full transparency. If the texture has a refractive index greater than one, it will be perceived only by the distortion of objects visible behind it.
- **For a perfectly reflective surface,** use no ambient, no diffuse, no specular, and no transparency, but set full reflectivity. The object is discernible only by the objects it reflects, which light is purely specular in nature (even though the specular fraction is off), and not from any diffuse component coming from its surface.

Also remember the three checkboxes at the bottom of the expert-level texture editing dialog box:

- **For metals,** use the "Link diffuse/specular colors" checkbox. Metallic surfaces reflect in the color of the metal, not the color of the light like other types of surfaces. Note that there must be something to reflect in the object to make it look metallic. Otherwise the surface will appear dark and flat. You can use an environment for realistic reflections of lights and darks, or build other objects in the model to reflect in the metallic surface.
- **To allow light to pass through a surface,** turn off the "Enable object shadow casting" checkbox. You do not have to make the surface transparent. This feature would be used with glowing

textures where a light source is positioned inside or behind the objects.

- **When using surface maps,** make sure the "Property values scale maps" check box is enabled. Otherwise, all surface maps are given full range.

The most important thing to remember with textures is that the settings for a given surface appearance are not just picked out of a hat. They must be thoughtfully measured and composed with intelligent application of the appropriate surface properties. There is no guarantee of realism; it is up to the designer to analyze how to simulate the desired surface and to supply a rationale for the resulting texture definition.

The second most important thing to realize is that the settings in the Surface Properties fields interact with each other in complex ways, and that there is no single set of "right" values. Additionally, the fields in a texture definition are not logically exclusive. It is possible to create a surface that is highly reflective, yet with a matte surface. Or think about the possibility of transparent wood — something that does not exist in reality, yet is easily described and rendered in Vision 3d.

In a more useful example, a texture may have a surface map giving the appearance of marble, a high smoothness factor, and a reflectivity that hints of a polished finish, but with just a touch of transparency thrown in to give the appearance of a thin sliver of stone with a translucent quality. The possibilities are limitless, but it does take experimentation and patience to learn the best techniques.

Surface maps — explanation of fields

Rendering objects with textures using only the Surface Properties fields produce images with uniform surfaces....which makes them look computer generated. This is in contrast to the majority of objects we see and handle in the world about us. The goal, then, should be to render images that look like they DIDN'T come from a computer. This is where surface maps can help.

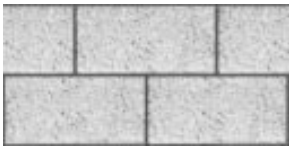
The conventional direction taken to enhance surface detail is to build complex surfaces with explicit geometric modeling. *This is the wrong technique!* Surface maps, in the form of PICT images, are more efficient, easier to create and offer an attractive alternative to building complex surfaces.

For example: It is impractical, if not impossible to create a realistic cinder block surface with explicit modeling. Further, there is no need to create the individual blocks of a wall either. Rather than creating a cinder block wall piece-by-piece in Vision 3d, you need only use digitized images (one for the color map, if needed, and one for the bump map) of a section of a real wall and map it to a single polygon object of appropriate size. In addition to being easier and more efficient, the model renders faster because there are fewer objects to process.

Cinderblock Wall

This wall is modeled from a cube primitive for efficiency (using the Scale tool to flatten it), then the cinderblock texture is applied from the front. The mapping style is set to cubic, with tiling to repeat the map so it covers the entire surface of the object.

Example of cinder-block surface map in texture applied to above wall



Most of the surface properties fields described in the previous section have a corresponding surface map field. They work in

concert when a map is present. The surface properties fields set the level or ceiling for each property. The map modulates the surface pixel-by-pixel within the range set. If no surface map is in place for a certain property, no modulation occurs and all areas of the surface appear uniformly at the level set by the Surface Properties field.

The properties that don't have a corresponding map field include Index of Refraction, First Smoothness, and Second Smoothness. Thus refraction and smoothness properties are, by circumstance, always global in their effect.

Color or Grayscale maps

Two of the mapping fields accept color images: Diffuse Color (a.k.a. the Color map seen in the primary texture editing dialog) and Specular Color. The rest of the surface map fields REQUIRE grayscale or B&W only.

NOTE: *Do not confuse the Diffuse Color and Specular Color maps, which use color maps, with the Diffuse and Specular maps, which use grayscale maps, or the Diffuse and Specular color fields, which are not map fields but global color settings. While their names are similar (because they all have effect on the reflected diffuse and specular light), each of these fields perform a different function.*

Color maps control the color of light reflecting from a surface based on the color of the pixel in the map at each point on a surface. The shading algorithm then uses those colors, along with the color, intensity, and direction of the light falling on an object to calculate the actual rendered color of each point on its surface.

Grayscale maps control the amount of reflected light for each type of surface property, again based on the gray level of the pixel in the map at each point on the surface.

Using Diffuse and Specular maps.

As noted, diffuse color maps are the main method of putting color images on objects. They are visible when diffuse light reflects off the surface. On the other hand, specular color maps allow you to have pixel-by-pixel color control of the specular light reflecting from surfaces.

Diffuse Color and Specular Color surface maps are usually created in a paint, drawing or image-processing application with any color palette — from one- to 24-bit. For realistic appearance, a color palette with at least 256 colors (8-bit) is recommended for color maps; a maximum of 256-grays is supported for grayscale.

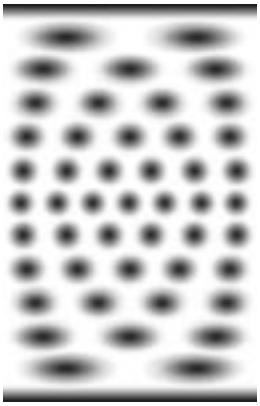
The depth at which color or grayscale surface maps are stored within Vision 3d is set at the time they are loaded into a texture. This can be different than the pixel depth at which they were saved from other applications, though you should be careful of adversely affecting the quality of the image when reducing the color depth. Setting a surface map's pixel depth is discussed later in this chapter.

USING BUMP, DIFFUSE, GLOW, SPECULAR, REFLECTIVITY, AND TRANSPARENCY MAPS

Pixel-by-pixel color control is not the only kind of information that can be mapped. Surface depressions or rises can also be simulated with surface maps using a technique called bump mapping.

In addition, a pixel-by-pixel description of a surface's reflectivity, transparency and glow characteristics can also be mapped to an object. These maps do not use colors to generate their effects, but work with the brightness values of grayscale images.

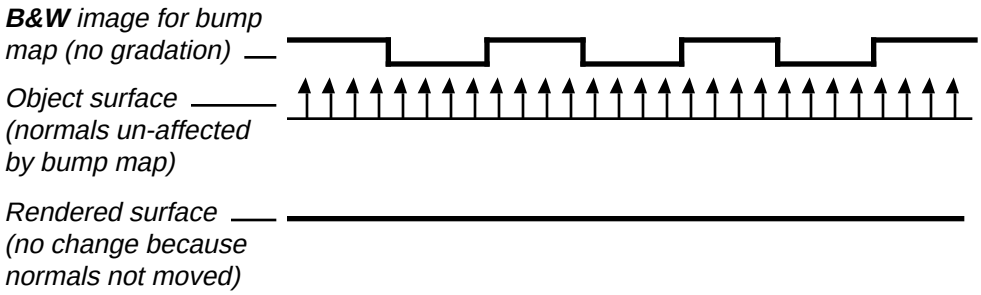
- **Bump Maps:** A bump map is a flat representation of an uneven surface which manipulates the interaction of light with a surface. In this manner, it fools the renderers into thinking the surface has highs and lows in it — the dark areas visually pushing the surface down; the white areas leaving it at normal height. Bump mapping avoids the complexity and overhead of having to model explicit geometry to create secondary surface variations or "bumps."

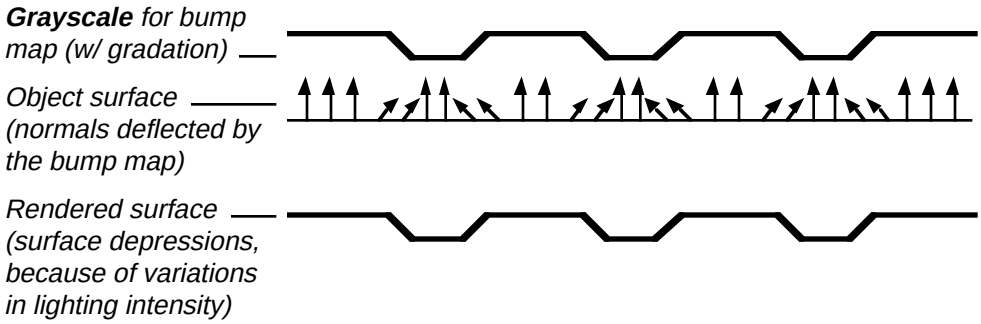


It is very acceptable for the majority of usages, even though the bump mapping illusion can break down when looking at an object's profile — it does not change the silhouette of an object. Bump mapping does not change the actual geometry of objects, but rather alters the direction the "surface normals" are pointing.

Because bump mapping works with surface normals, the map requires transitional grays between the white (high) and black (low) areas of the map to determine how deep the depression appears. This can be introduced in the map by simply blurring

the image once or twice in an image processing application before loading it into the Bump map field of a texture.


B&W image for bump map (no gradation)
Object surface (normals un-affected by bump map)
Rendered surface (no change because normals not moved)


Grayscale for bump map (w/ gradation)
Object surface (normals deflected by the bump map)
Rendered surface (surface depressions, because of variations in lighting intensity)

In technical terms, these transitional grays provide gradient vectors to "perturb" or bend the surface normals. A sudden change from white to black, with no gradient between them, will not register as a surface depression.

- **Diffuse Maps:** A diffuse map controls how much diffuse light is reflected from an object's surface, based on the pattern in the map. Black areas prohibit any light from being reflected; Dark gray areas suppress diffuse light according to the level of gray; white areas allow full diffuse reflection (at the ceiling specified in the Diffuse Fraction surface property field).

  Again, do not confuse the function of a diffuse map with that of a diffuse color map. Diffuse maps control the AMOUNT of diffuse light reflected, while diffuse color maps determine the COLOR of the light being reflected.

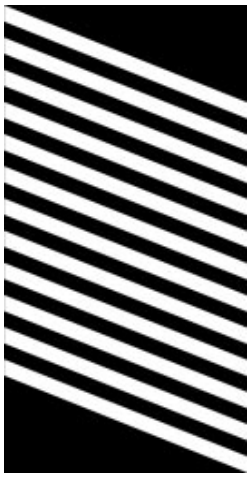
- **Glow Maps:** The glow map controls where (and how much) glow occurs on an object's surface, based on the pattern in the map. As with other grayscale maps, glow maps use whites to allow the highest amount of glow, while grays lessen the glow, and blacks completely suppress it. The maximum glow is determined by the setting of the Glow Factor surface properties field.

  Vision's "glow" feature is provided to simulate the appearance of items illuminated from within or behind. A glow map lets you control where the glow occurs on a surface. Without a glow map, the entire surface glows at the same level.

  As noted earlier in this chapter, objects with a glow factor in their texture are not light sources and will not illuminate adjacent objects.
- **Specular Maps:** Using a specular map in combination with its corresponding specular fraction field allows you to model surfaces that vary in their gloss or shininess. When manipulated properly, the interaction of the specular and reflectivity maps can provide important visual clues about the composition of objects and accurately simulate surfaces ranging from metals like gold, aluminum or brass, to plastics, glass, paper and paint.
- **Reflectivity Maps:** Reflectivity mapping uses a grayscale image to vary the level of reflectivity across an object's surface, based on the pattern in the image. Again, the Reflectivity field in the Surface Properties determines the maximum reflectivity and scales the range of the reflectivity map. Whites in the image correspond to the most reflective areas of the surface. Grays lessen the reflectivity, and blacks produce no reflectivity at all.

  A reflective surface is defined using the global setting of the Surface Properties reflectivity field. However, few surfaces (except perhaps mirrors) are perfectly uniform in their reflectivity. By using a surface map to define the pixel-by-pixel reflectivity values, you can more realistically model polished or shiny objects.
- **Transparency Maps:** Use transparency maps to define areas on a surface that are transparent. This mapping type, more than any other, can give the illusion of geometry different than the actual object. You can make holes through objects, eliminate parts of an object, add features that do not exist in the object. Examples of transparency mapping include doorways, windows or other

openings in walls, windows in an airplane fuselage, holes through the surface of any object, and mapping human outlines to simple rectangular objects.



Grayscale maps also allow you to simulate an object's translucent patterns or irregularities. As with other surface mapping fields using a grayscale map, the white areas of the map correspond to the highest transparency level (set by the ceiling value in the Transparency field), grays produce translucence and blacks result in opacity.

Using multiple surface maps

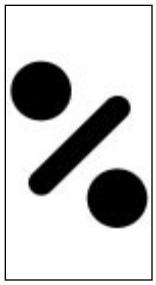
It is very common to use the same grayscale image, or an inverse of it, in several map fields within the same texture. Typically, the same surface details are present in each of the various maps, and are often created from the same master image. That way, they are guaranteed to register with each other correctly when rendered.

For example, suppose you have a lathed object that you want to make partly transparent, i.e. — appear as if there is a hole through the surface. Now, you could try to model the object so its geometry actually did have a hole. But, sometimes the tools and modeling extensions in Vision 3d just aren't able to do what you want. What then? *Where the modeling tools leave off is where the power of surface mapping begins.*

When using mapping to make holes, there is one very important concept to be remembered. Those areas of the surface you want to make a hole must also be suppressed with respect to the ambient, diffuse, and specular light that would normally reflect

from that area. Otherwise the hole will not be believable. This requires using multiple surface maps to control the various surface properties in specific ways.

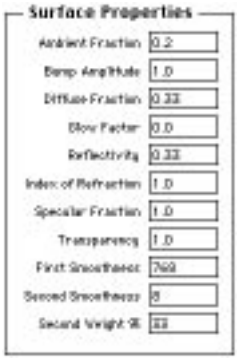
A transparency map (at right), is combined with ambient, diffuse, specular, and reflectivity maps



Ambient, Diffuse, Specular, and Reflectivity map

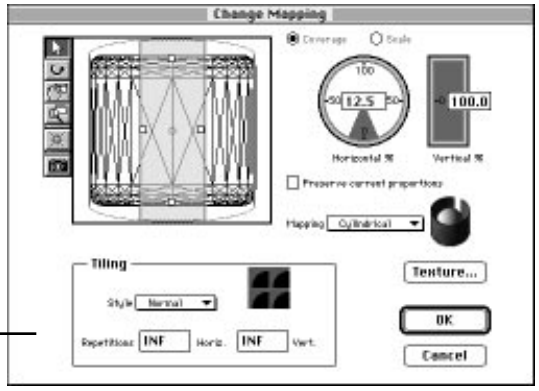
Transp. map

Then, the Surface Properties fields are set as shown (far right)

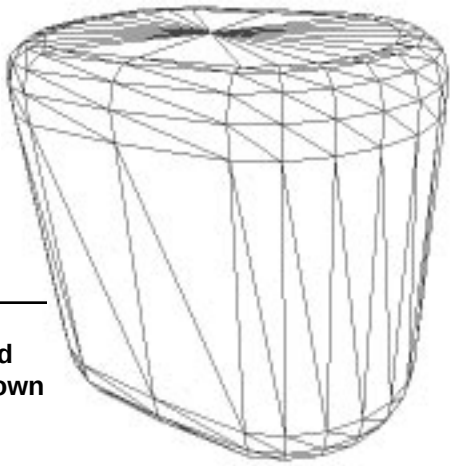


The texture is applied to a stretched, rounded cylinder.

In the Change Mapping dialog, the texture is set to wrap in a cylindrical pattern, with normal tiling



Wireframe object



The rendered object is shown at far right!

Thus, from the above illustration, you can see how the ambient, diffuse, specular, and reflectivity surface maps are the same image. In this case, because the surface has been given some amount of reflectivity, a reflectivity map is also required to suppress visible reflections on the surface where there are supposed to be holes.

Note how the four light-handling fields all use the same map. On the other hand, the transparency map must be an inverse of the other maps. Even though the transparency map is inverted from the other map, each of these mapping fields work exactly the same way, i.e. — white allows the specific property for that field, black suppresses it, and gray modulates some amount in between depending on the level(s) of gray. In the case of the transparency map, white is the area of the surface that is to be transparent. But note that this is the same area on the surface that the ambient, diffuse, specular, and reflectivity maps must be black to suppress reflected light, since no surface is supposed to be where there is a hole.

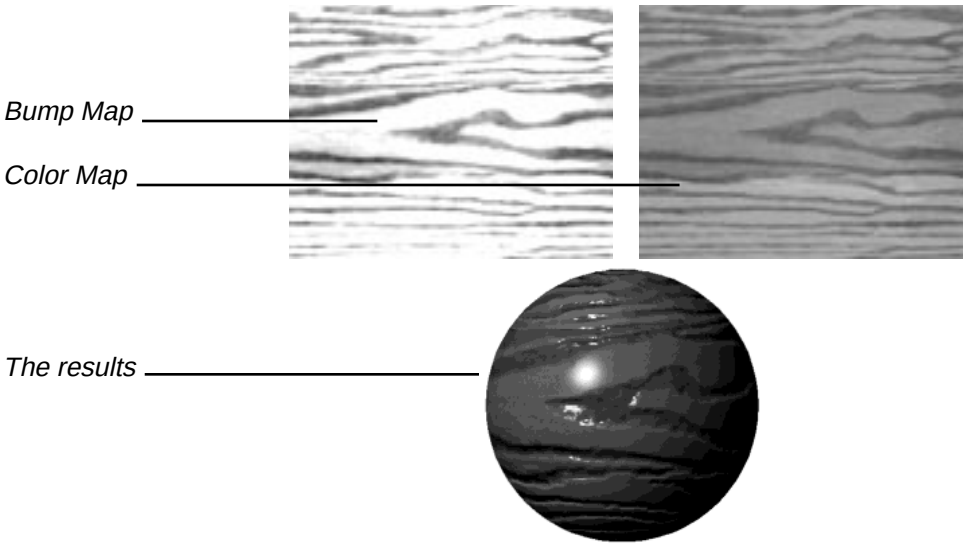
This is a typical use for multiple surface maps. When transparency mapping is used in conjunction with maps in the other fields, the object responds to the light sources as if there are actual holes in the surface…and the rendered image is totally believable.

Registering multiple maps

You should insure that all the maps you intend to use together in a texture will align or register with each other, and work correctly with each other, before inserting them into a Vision 3d texture. No controls are provided to independently position surface maps once they are within a texture, so the details or patterns within the maps must be registered with each other during their creation in the paint or image processing application used to make them.

Thus, when creating your own multiple surface maps for use in a texture, they should usually be the same size so that the areas of detail in each match up. If the texture is to be repeated to cover the surface, the tiled repetitions will match properly.

For example, this would be necessary if you wish the roughness of a bump map to match the grain in a wood color map.



This uniformity of map handling is important to understand. When applying a texture with multiple surface maps, they are manipulated as one, i.e. — mapping style, scaling, rotation, etc. All maps are aligned or centered on the texture origin point of the object and are oriented according to the active view in which the texture was applied. All maps in the texture are scaled in unison using the Change Mapping command in the Textures menu. All maps use the same mapping type. Changing a texture's orientation on an object (by using the Control key with the object rotate tool), rotates all maps together.

USING MAPS OF DIFFERENT SIZE

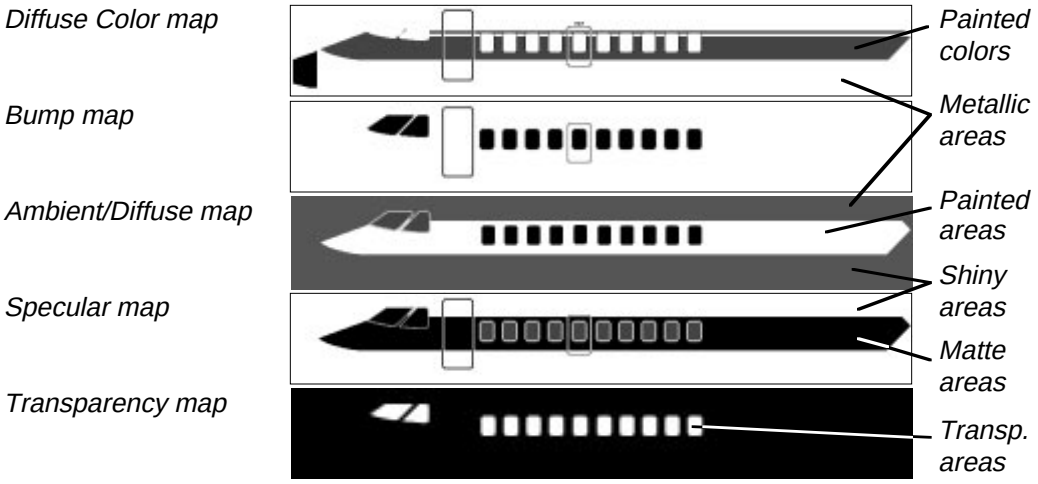
If required, you may use maps of different sizes in the same texture definition. This could occur when you wished to repeat a small pattern as a bump map, but had a larger color map meant to fit the object without tiling. When the tiling style is enabled, maps of different sizes repeat as needed across an object's surface.

Making composite maps

As mentioned earlier in this chapter, only one texture can be applied to an object. At first thought, this may seem to be a significant obstacle to effectively using the mapping capability in Vision 3d. In actual practice, with appropriate planning, this is a minor inconvenience.

It does require the use of "composite" maps, however. Composite mapping is the process of putting the properties of different surfaces in the same set of maps. Using this technique, you can have a colorfully-painted area, a transparent area, a highly reflective area, and even a metallic area next to each other on the same object.

It is just a matter of identifying how you will be applying the texture, i.e. — the direction of application, the mapping style, and tiling method (if any). Then, you must decide what surfaces you want and where they need to be positioned on the various maps as they wrap onto the object's surface geometry. The specific maps required are determined by the surfaces you want, and how you plan to "communicate" the appropriate values for each surface property. This is shown in the following illustrations of the fuselage composite maps used to render the image in the previous illustration.



Of course, there is usually more than one way to build composite maps. Start with something easy — perhaps a simple rectangle — and experiment with the concept. Composite maps works incredibly well; you just have to gain some experience to tap into their power.

USING WIREFRAME IMAGES AS TEMPLATES

How do you know where to put the components or elements in your maps? That's easy! Just get a template of the object on which you want to apply the surface maps. Save or screen-capture one or more wireframe PICT images of the object directly from the Vision 3d modeling windows (from the direction you plan to apply the texture). Then, open these wireframe images as templates in the application where you will create the maps. That way, you can "paint" right over the wireframe (erasing to white the areas you don't cover up) and you know the maps will be the right proportions and the elements within the maps will be positioned correctly on the object.

If you need to make the maps larger to improve their resolution — because you will be rendering at higher resolution — you can increase the size of the template an appropriate amount BEFORE you start drawing over it. The wireframe template may become fuzzy or pixelated, but that doesn't matter because you'll cover it up or erase it from the maps anyway.

NOTE: *The amount of resolution you need to build into maps for higher resolution renderings is discussed later in this chapter. Please refer to this area in the manual AND UNDERSTAND IT before beginning a composite map project. You may find you do not need as much as you think.*

AN EXAMPLE OF MAKING/USING COMPOSITE MAPS

Suppose you model a bottle that you want to appear made of glass except where two labels are positioned on the front and back. By compositing some appropriate maps, you will be able to make the bottle transparent everywhere except where the labels are positioned, which must also have a high amount of diffuse light reflecting from them for appropriate brightness.

Using the one texture / one object rule, if would not seem possible to have part of the bottle transparent and the part where the label is be opaque. Yet the following illustration is proof that it is not only possible, but it is a very useful and powerful technique!

The question is not whether it can be done,
but rather where do you begin?

Looking at the lathed bottle in the model window from the Front view (in wireframe, orthographic display), choose the Save As command. Switch to PICT as the save format, enter an appropriate name, and click on the OK button. Vision 3d will save a draw-PICT

file of the active view. Saving a PICT directly from the model window is an excellent way to generate the templates you need. You may actually find you prefer this method to screen captures.

Alternately, you can use a screen capture to obtain the wireframe template. If you do not have a commercial screen capture extension that allows you to drag a marquee around the portion of the screen that you wish to capture, you can use the built-in screen capture feature in System 7. However, this screen capture mechanism saves a PICT file of the entire desktop, so you will have to crop in on just the bottle template afterward. To initiate a System 7 screen capture, hold down the Command and Shift keys, then push the three (3) key. The Operating System responds with a camera shutter sound. After a brief pause, a PICT file titled “Picture 1” appears in the root window of your machine’s startup hard-disk drive. Each subsequent screen capture is named Picture 2, Picture 3, etc. For more information, please refer to the Operating System’s documentation concerning this feature.

Planning the mapping

For this example, you will use the template to make a double-width map that wraps completely around the bottle. That way, you can have different labels on the front and back of the bottle, as well as any other surface detail you want anywhere on the bottle’s surface…because you are using composite maps and you are in control of the entire surface!

Most bottles are constructed with the lathe modeler. This implies that the mapping type to be used with a surface texture applied to the bottle would be cylindrical. The first thing you must decide is the dimensions of your bottle. It is essential for setting the dimensions of the surface maps you will make .

Setting map size

So, assume you make your bottle 4” in diameter and 6” high. (This discussion assumes you know how to draw a single-walled bottle template and lathe it into a 3-D object. Please refer to Chapter 3 if you need help in this task.)

Using simple junior high math, you know that the circumference of the bottle is found with the formula $C=2\pi r$ (where "r" is the radius) or its derivative $C=\pi d$ (where "d" is the diameter). Thus, for a 4" diameter bottle, the map would be 6" high by 12.567" wide. This will wrap completely around the bottle without distortion.

Actually, the only thing you're really interested in is the proportion or aspect ratio of the map. If the width is too long or too short for the height, the map will distort when wrapped around the bottle. The actual size isn't as important as the proportions, but by working in actual size, the scale and position of elements in the maps may be easier to visualize.

Making the maps

You can use a paint program, but it may be difficult to work with all the elements. It is usually better to use a drawing program to construct the label, then convert it to a PICT image for mapping. Adobe© Illustrator® works well as the drawing application, with Photoshop™ doing the conversion to PICT. Of course, Illustrator is PostScript-based. Canvas™ or SuperPaint™ are excellent QuickDraw-based drawing applications that can save PICT files directly. You may wish to build all of the maps separately in the drawing software, then convert them to PICT format, or you may wish to

build only the color map then make the other grayscale maps based on the PICT color map.

Color map

The first map you want to make is the color map (a.k.a. Diffuse Color map in the expert-level texture editing dialog). Here you will define the position and contents of the labels. Again, if you want to build this map in a drawing program, just make sure you can convert it to a PICT afterward. Of course, you want to crop the PICT document to the 6" high by 12.567" wide dimensions calculated earlier.

As you can see in the above illustration, there will be two labels — front and back. There are also two little logos that will show on the neck of the bottle. Notice that the rest of the color map is white. This is the area that corresponds to the area you want to be transparent. If you want the glass in your bottle to have a color, this would be where you define that color. For best reproduction the color map is usually made in 32-bit color. However, if you can switch it to 8-bit color, you will be able to save a substantial amount of memory. Save this map in PICT format, with a name like "Bottle - color map."

Transparency map

Once you have the color map saved, you can use it as a pattern to make the transparency map. In this example, all you need to do is select the areas you have defined as the labels and fill them with black.

The label areas are nice rectangles, so the selection and fill process is fairly straightforward. The concept behind the transparency map is that the white parts allow that area of the bottle's surface to be transparent, while the black prohibits transparency. Since this is a grayscale map, in other projects you can also use levels of gray to set different levels of transparency on a surface.

Before saving this map, convert it to grayscale. Save it with a name like "Bottle - transparency." When actually loaded in Vision 3d, this same map will also be used in the Specular and Reflectivity map fields as well.

Ambient and Diffuse maps

Next, you need to make a map that designates which areas of the bottle's surface will reflect diffuse light. A little thought about this should give you an understanding that the labels should reflect the highest amount of diffuse light. On the other hand, you do not want a lot of diffuse light reflecting from the transparent area of the surface, because that would obscure the transparency. It would have the effect of making the transparent area milky or even opaque.

Remember, each of the mapping fields that use grayscale maps use the same principle: White allows that field's surface property, black prohibits it, and grays modulate the property depending on the level of gray.

As you look at the transparency map, you can see that by simply inverting it, the white parts of the map would nicely correspond to the labels and the black part would suppress the diffuse light where it will be transparent.

If the transparent area is pure black, no diffuse light will be reflected from the surface of the glass, making it completely transparent. In actual practice, it is more realistic to make the glass area of this map about 80% black (or 20% white, whichever way you want to look at it). So, fill the black area with an 80% black — making sure the white parts are not encroached upon. You want that area of the surface to reflect ALL of the diffuse light.

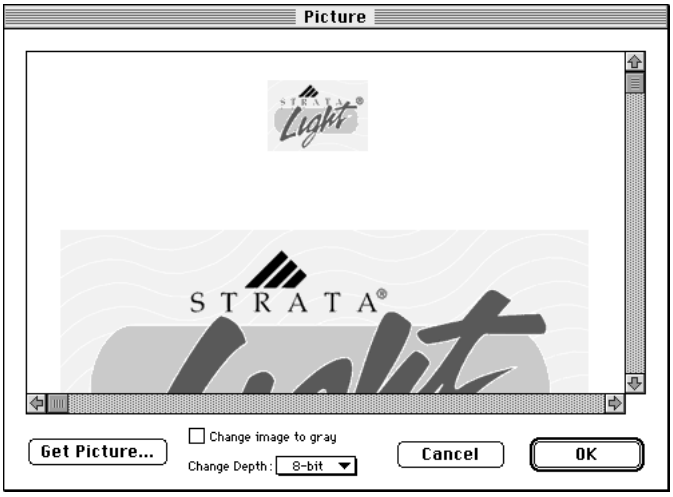
When this map is placed in the ambient and diffuse map fields, only 20% of the diffuse light striking the glass area will reflect, providing just a hint of the surface.

Save this map with a name like “Bottle-amb / diff.”

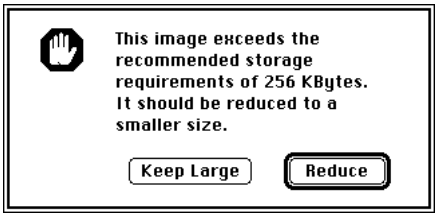
Putting it together

Now it’s time to return to the Vision 3d model of your bottle. With the bottle selected, click on the New button in the Textures palette. Alternately, you can click on the New button in the Textures Browser dialog, but that will not automatically apply the texture to the selected object when completed, as the New button will.

In the primary texture editing dialog, click on the Color Map button. The Picture dialog box appears. Click on the Get Picture button and load the Bottle - color map file. You should see your label masterpiece loaded into this dialog.



Notice the controls at the bottom of this dialog. The color depth defaults to 32-bit, even if the PICT file was saved in 8-bit (as suggested in this case). If you are using an 8-bit image for the color map, switch the setting in the Change Depth pull-down menu to 8-bit, then click on the OK button. The color map draws in miniature on the texture editing dialog.


This image exceeds the recommended storage requirements of 256 KBytes. It should be reduced to a smaller size.
Keep Large
Reduce

You will see an alert saying this image is larger than the suggested 256K. For this project, you want this map and all the other maps to remain the size you made them. Choose the “Keep Large” option. *THE “REDUCE” BUTTON IS NOT THE CORRECT CHOICE HERE!* When using multiple maps that you have created to register with each other, reducing an image changes that map’s dimensions by whatever amount necessary to bring the image under the 256K memory-storage threshold; causing the planned registration between multiple maps to fail! (Other projects may not have the same requirements as this example and it would be appropriate to reduce them to save memory.)

Expert level

Next, click on the Expert button to enter the expert-level of the texture. This second dialog allows you to load the specialty maps and set the surface properties to match the maps you have used.

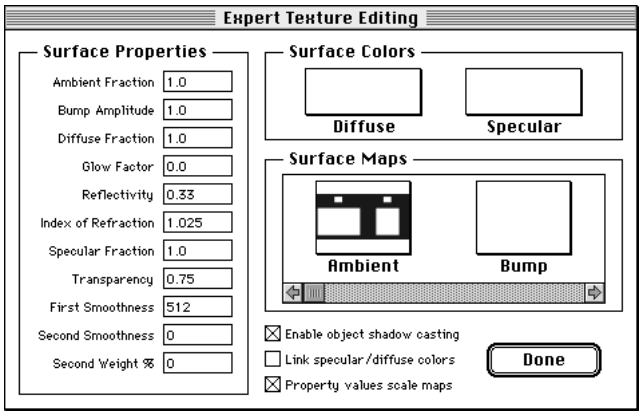
Click on the Ambient map rectangle. You will see the same Picture dialog used previously. Load the Bottle-amb / diff file, then click on the OK button. Note that the default color depth for these map fields is 8-bit grayscale.

Repeat the process for the Diffuse map, using the same file. Note that even though they are in the same texture, each map field must have its own image loaded into it.

Now, scroll the Surface Maps area of this dialog to the right to show and insert maps into the Reflectivity, Specular, and Transparency fields. This time, use the "Bottle-transparency" image for these three fields. Why? Because glass is glossy while labels are not usually shiny. Not only do you want to designate where the bottle will be transparent (the white part of the image), but you need to control the areas where the surface of the bottle will be reflective and shiny. In this case, they happen to be the same areas. Thus, you can use the same map for these surface properties.

Setting surface properties

With the maps in place, you must now set the ceiling values for the surface properties under which the maps are to work. Obviously, if you have installed a transparency map then you must turn on transparency. If the transparency surface property is set at zero, then the white part of the transparency map will still be no transparency.



Set the various surface properties to the values shown in the above illustration of the expert-level texture editing dialog. The rationale for the numbers should be fairly straightforward. Take time to try an understand them. Click on the Done button to close the expert dialog. Enter “Bottle Label” as the name of the texture and click on the OK button to accept the texture.

The setting in the Index of Refraction field deserves special mention. The usual setting for glass is 1.5 or 1.6. You could use this value, but the rendered image would give the appearance of a bottle filled with a very thick liquid, because the refraction is so strong. There would be an exaggerated distortion of the back label through the glass of the bottle. This may be the effect you want. Others may think the bottle looks more realistic with just a hint of refraction, like you would see from a thin-walled, empty glass bottle. This is an area in which you can experiment during the preview renderings.

Applying the texture

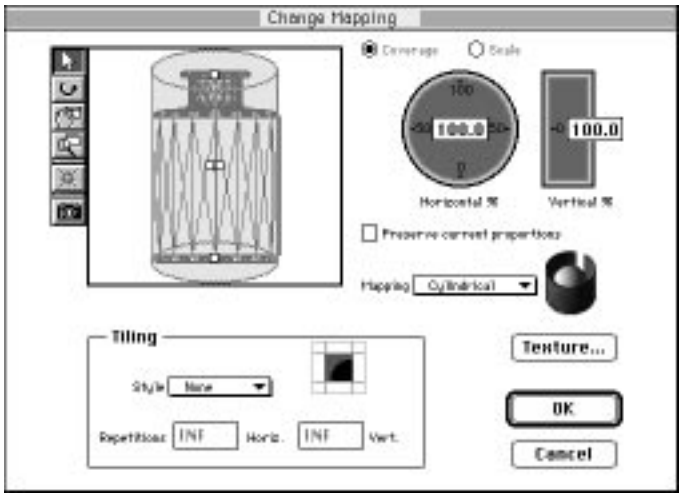
As with any surface texture, the direction of application determines the orientation of the maps within it on the object’s surface. Remember that the bottle’s maps were created with the front label on the left side and the back label on the right. That means the center of the label is really meant to be the side of the bottle. A little mental visualization should help you understand the relationship of the maps to the bottle.

Thus, you want to apply this texture from the right side of the bottle, so the front label appears on the “front” of the bottle. Switch to the Right view, then choose the Bottle Label texture on the Textures palette. Click on the Apply button to apply this texture to the bottle from its right side. Alternately, you could select the texture’s preview image then drag it into the right view of the model window and drop it on the bottle.

Setting the mapping parameters

Next, you must set the mapping parameters through the Change Mapping dialog. Open this dialog by clicking on the Change Mapping button — the one that looks like the following illustration. This allows you to set how the bottle label texture is mapped on the bottle.

Obviously, the first thing to set is the mapping type. Click on the Mapping pull-down menu and choose “Cylindrical” mapping. Next, set the tiling style to “none.” This is not absolutely necessary, since the texture is designed to completely cover the bottle’s surface without tiling, but turning it off allows you to see easier where it may not be covering (in case off an error in the dimensions).



Finally, you want to set the scale of the surface maps. Remember back to the beginning of this project. Using the Transform dialog, you made sure the bottle was 6” high and 4” in diameter, then calculated that the width of the label maps would need to be 12.567” to have the appropriate proportions. Thus, it is a simple matter to set the vertical coverage to 100.0 (meaning 100% coverage) and the horizontal coverage to 100.0 (meaning 100% around). Click on the OK button to accept these mapping parameters.

Rendering the bottle

Everything is now in place to render the bottle. You may want to make a couple of preview renderings to verify the mapping is the correct orientation and scale. You may also want to adjust the lighting and put some other objects in the scene to add to the realism. Also, be sure to check the Index of Refraction for your personal tastes and what you want to communicate about the thickness of the glass.

Of course, Raytracing is strongly recommended for realism in the rendered image. Only raytracing will accurately calculate the

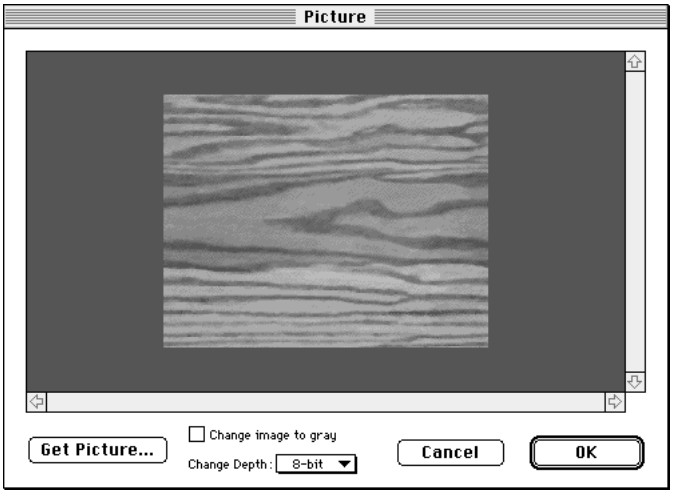
refraction and transparency. In the rendered image shown earlier in this example, an environment was also loaded into the model to reflect off the glass and further enhance the glass surface's appearance. Again, note that the reflections only occur on the glass, not on the label. That's because of the Reflectivity and Specular maps you loaded into the texture.

In the example used here, the bottle is lathed from a simple single-wall template. You may want to construct your bottle with a double-wall template so you can have stronger sense of glass thickness. That would also facilitate modeling something inside the bottle while still retaining the glass refraction.

Loading and removing Maps

Surface maps may be added or removed from a texture at any time. Using the Texture palette's Edit button to open a texture for editing, then moving through the various dialogs as needed, you can insert or delete previously created surface maps quite easily. Each type of surface map is stored in a separate field within the expert-level texture editing dialog.

To load or remove a surface map, click on the appropriate rectangle in the expert dialog box for the map you are loading. A Picture dialog box appears. This is the dialog where surface maps are loaded into and/or deleted from a texture.



Click on the “Get Picture” button at the bottom of the dialog and select the desired PICT file in the directory dialog that is displayed. You may also copy and paste images to and from the Scrapbook in this dialog.

To delete a map, open this dialog for the map you wish to remove and push the Delete key or choose the Delete command in the Edit menu. You may also remove it using the Cut command from the Edit menu.

Solid textures

As indicated earlier in this chapter, solid textures are volumetric in nature. That means, their patterns are defined as a three-dimensional volume out of which objects appear to be carved.

While solid textures allow you to set the full suite of surface properties, just as described in the surface texture discussion earlier in this chapter, they do not have the extended pixel-by-pixel control afforded by surface maps. Specifically, solid textures do not offer bump mapping.

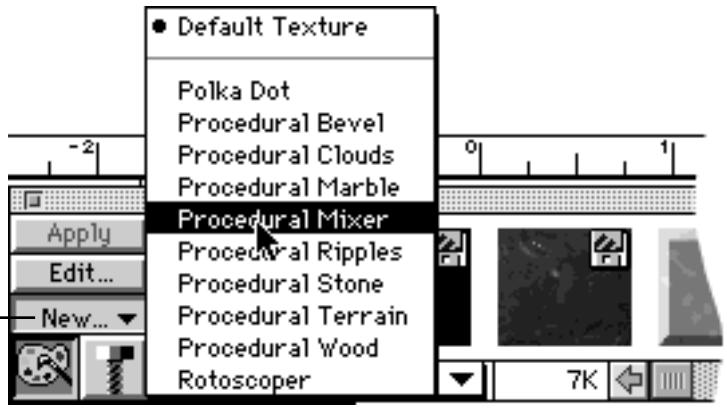
Surface property settings in a solid texture are global in their effect. The pattern contained in a solid texture is roughly analogous to the diffuse color map in a surface texture, except that it is volumetric in the way it propagates the pattern and it is infinite in size.

Thus, although the scale of the pattern is adjustable, no mapping or tiling settings are required. When using the Change Mapping dialog to adjust the scale, position, and orientation of a solid texture on an object, notice how the irrelevant fields of the dialog are dimmed. Additionally, a Scale field is included in each solid texture’s editing dialogs to adjust the pattern density…something not found in the dialogs for a surface texture.

Alternately, you can use the Object Scale tool in combination with the Control key to scale the texture applied to an object instead of the object in the model view. This works for both texture types, and is discussed in *Chapter 3: Modeling Objects*.

Several solid textures are shipped with the software and can be loaded into models, modified, and copied between models as needed. Additionally, you may make your own solid textures, based on the available texture extensions installed in the Extensions folder (within the StrataVision 3d application folder).

Normally, clicking on the New button gives you the standard dialog for a surface-type texture. If you wish to create a new solid texture, hold down the mouse button until the texture extensions pop-up menu appears.



Hold down mouse button when selecting New to choose from texture extensions.

You are presented with a list of the available texture extensions (upon which solid textures are based). Choosing a texture extension from the menu opens the appropriate texture-editing dialog connected to that texture extension.

Texture extensions explained

As introduced at the first of this chapter, each solid texture requires a “texture extension” from which to generate its color pattern. Because of their complexity, the texture extension files must be compiled and assembled externally with special segments of code that tell the renderers in Vision 3d how the pattern it describes will propagate volumetrically through an object when rendered. However, any number of solid textures can be created (and saved) from one texture extension — variations on a theme, if you will.

Sometimes, these texture extension files are also called “procedure maps.” Each procedure map / texture extension file carries its own dialog boxes for choosing colors, setting scale, and generally describing how a solid texture using that map should appear when used in models.

It is important that you understand if a solid texture's texture extension file is not found in the Extensions folder within the StrataVision 3d application folder, when you try to re-open a model containing a solid texture, it cannot be accessed. In fact the model cannot even be opened…that is how important the texture extension is!

A solid texture can be saved as a separate file, which is then available for loading into other models as desired. But the texture extension that supports that solid texture must also be present in the Extensions folder for it to work.

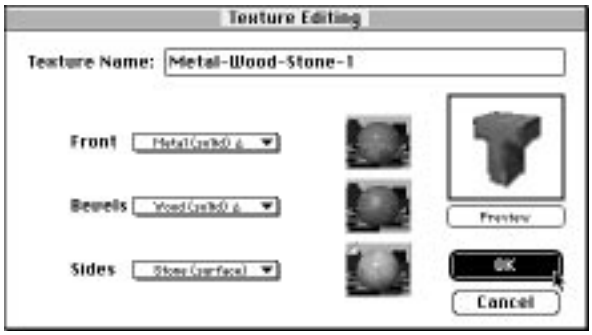
Texture extensions, and the solid textures that use them are similar to the operation of PostScript® fonts, i.e. — each font must have two parts to work properly: The bitmap / screen font and the Type 1 or Type 3 outline support document. You may use a screen font in a document, when it is installed in your system. But to get the font-smoothing benefits and to re-open the document or open it on another machine, the postscript portion of the font must also be present.

Solid textures work in a similar manner.

SPECIAL SOLID TEXTURE #1: BEVEL TEXTURE

To use this extension, choose Procedural Bevel in the New Texture button's pull-down menu.

Choose currently loaded textures for the front, bevels and sides.



The Bevel texture allows you to specify three specific textures — either surface or solid type — that will be applied to the front, bevel, and sides of an object. A bevel texture can be applied to any type of

object, although extrusions usually work best with this type of texture.

You can choose from textures currently loaded in the model. Only textures currently loaded in the model are available in the Procedural Bevel extension's pull-down menus. Textures CANNOT be chosen from the texture palette while creating a new bevel texture; they must be loaded into the model FIRST.

To preview the newly created bevel texture, click on the Preview button. When satisfied with the bevel texture choices, click on the OK button. You can now apply the texture as usual. Remember, the bevel textures you create will be direction sensitive just like any other texture. That means you must take care to apply them from the appropriate direction to get the effect you want, i.e. — front texture applied on the faces that are parallel to the active view, side texture applied on the faces that are at right angle to the active view, and the chosen bevel texture to anything in between.

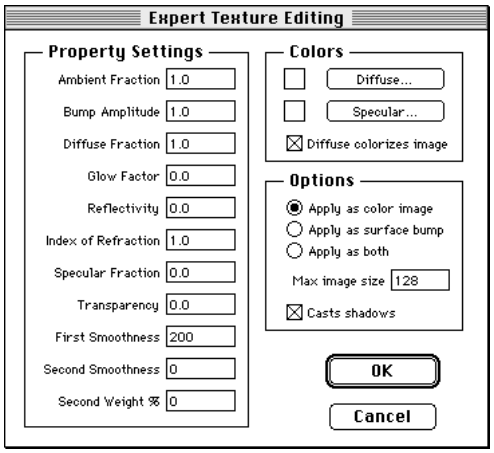
SPECIAL SOLID TEXTURE #2: ROTOSCOPER TEXTURE

The Rotoscoper extension allows you to use a QuickTime movies or PICS files as maps that can be applied to objects. As with other solid textures, the Rotoscoper texture is accessed by choosing it from the New Texture button's pull-down menu. While a Rotoscoper-based texture is considered a solid texture, the animation files that it uses are obviously surface only.

This texture allows you to "play" an animation file as a surface map on objects as part of a rendered animation. The texture dialog provides fields for setting play speed, as well as when to actually start playing the chosen animation within the larger animation. Once a rotoscoped animation map is applied, you can scale it to fit the

object as you would any surface-type texture through the Change Mapping dialog.

As with every other texture, there is an expert-level secondary dialog with fields for setting the general surfaces properties. From discussions earlier in this chapter, you should already be familiar with these fields. Additionally, there are some special rotoscoping options for setting the maximum frame size of the animation being used and whether the frame will be applied as a color image, a bump map, or both.



Uses for the RotoScoper may include playing digitized video clips on the screen of a television set during a walk through of the room or applying a person's face onto a model of a human head and animating the facial gestures to smile, wink, talk, etc.

If you find that the frame size is too small to fit the object, as evidenced by a pixelized appearance to the animation when rendered, you can increase the maximum frame size field. This field defaults to 128 pixels square to conserve memory, but has no upper limit. This means larger frames of the animation are sampled down to a maximum of 128 pixels of whichever dimension is larger. The other dimension is also 128 pixels or smaller. If you increase the maximum frame size, you must reload the animation file into the Rotoscoper to force a resampling of the frames at the larger size.

Handling textures in a model

Textures created or loaded into a model become part of that model. Saving the model also saves its textures as part of the file. This is true whether or not a texture has been applied to any objects in the model space.

Creating new textures

As discussed at the beginning of this chapter, any number of textures can be created as part of a model. Choosing New Texture button on the Textures palette opens the default surface texture editing dialog boxes. Holding down the mouse button while clicking on this button allows you to choose a texture extension from the pull-down menu upon which to create a solid texture instead. A full explanation of this button is found later in this chapter.

Loading textures

Textures may be loaded into a model as needed. It is an easy way to transfer work done previously in other projects to the current project. Once you have acquired a repertoire of texture files, either from commercial libraries or your own work, this feature lets you assemble pre-built elements into a complex model very quickly.

A distinction should be made between loading a texture and applying a texture. When a texture is loaded, it is given storage space in the model, but it is not always automatically applied. Applying a texture is the act of putting it on an object or shape instance.

As mentioned at the beginning of this chapter (and reiterated here for emphasis), the active view from which a texture is applied determines the orientation of the texture. This is especially important with regards to surface maps. If the orientation is not as desired, you

can either rotate the texture on the object or re-apply it again from a different view orientation.

Surface textures with maps, as well as solid textures, can use the Change Mapping dialog to adjust how the texture fits on the object(s).

HOW TO LOAD

There are five ways to load a texture into an open model:

1. **Double-click on its preview image in the Textures palette.** This loads the texture only. No application or editing of the texture is attempted.
2. **Select a texture on the palette and click on the Apply button.** This loads and applies the texture to the selected object(s) in the active view.
3. **Select a texture on the palette and click on the Edit button.** This loads the texture and opens its dialog box for editing. No application is attempted.
4. **Clicking and dragging the preview image of a selected texture from the Textures palette to an object in the active view.** This loads and applies the texture into the active view. This is referred to as a "drag and drop" operation.
5. **Choose the Browse Textures command in the modeling menu and click on the Load button in its dialog.** This is a variation on method #1.

The method you choose depends on what you want to do with that texture. The first four methods assume the texture to be loaded is represented on the Textures palette. The last method is for textures that are located somewhere other than in the Textures folder (within the StrataVision 3d application folder).

The loading of textures contained on the Textures palette is further explained under the Textures palette discussion later in this chapter.

As mentioned, you can also load textures into the active model from any source through the Load button in the textures browser dialog. This feature is explained further under a discussion of the Browse Textures command later in this chapter.

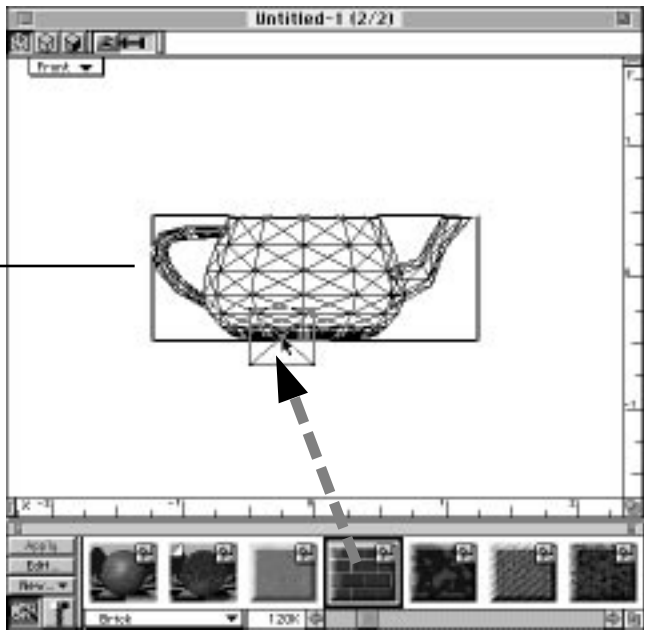
"Drag and drop" explained

As mentioned in method #4, you can click and drag a texture's preview image to an object in the modeling window to load and apply the texture. This is referred to as "Drag & Drop." This feature is available with any operating system. It is a feature of the StrataVision 3d application, and does not rely on System 7 for support.

Textures palette "Drag & Drop"

Click and drag the chosen texture onto an object in the active view

The texture is automatically loaded into the active model and applied on the object where you drop its bounding box.



A small marquee appears representing the texture as it is being dragged. As the texture marquee is positioned over an object, a marquee appears around it to indicate the connection is made. The object does not have to be previously selected. Upon releasing the mouse button, the texture is loaded into the model and applied from the direction of the active view on the object where you drop the marquee.

Note that the marquee you see during a drag and drop operation does NOT represent the size of the map(s) in the texture being applied. You will still need to use the Change Mapping dialog to make adjustments in a texture's mapping parameters, as with any other method of texture application.

Applying textures

Any texture can be applied to any object. The suitability of using a particular texture on a specific object is left for you to decide; you have full creative license. At first, you will probably use the textures provided with the Vision 3d software package. Later, you may begin to explore making your own textures, experimenting with the surface property settings for different effects, then adding a color map or a bump map, as well as other advanced mapping techniques, as you gain experience.

The work of constructing the object and creating the texture culminate when the two are brought together. At this point, you have a pretty good idea what you want the rendered image to look like. It is truly an exciting time when your project comes together and the rendered image actually looks just like what you saw in your mind.

Application methods

A texture can be applied to one or more selected objects at a time. There are two methods available to apply a texture:

- **Using the Apply button on the Textures palette.**
- **Using the "Drag & Drop" feature, also on the Textures palette.**

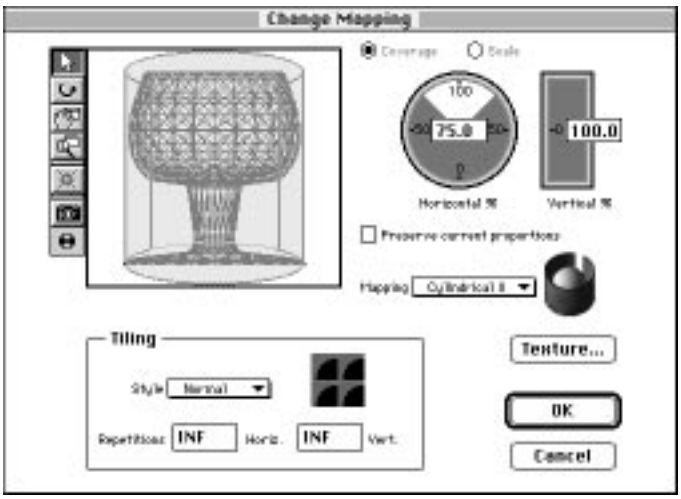
These methods work with any texture represented on the Textures palette, whether in the model or not. Applying a texture that is not in the model, but represented on the palette, will automatically load the texture.

Applying a texture is just the beginning. The balance of this discussion is concerned with the other considerations in texture application, including application direction, mapping type, scale, and tiling. As you can see, there are several variables that allow you to "fine tune" the way the texture is applied to the object.

Change Mapping dialog

While a texture can be applied to more than one object, only one texture can be applied to one object. How that texture is applied to each of those objects can be customized in the Change Mapping dialog box. That way, the same texture may be customized to fit different objects in different ways.

The Change Mapping dialog is opened via the Change Mapping button on the Button bar. If no texture is applied to a selected object, to a shape instance at this level, or if more than one object is selected, the button is dimmed. The Change Mapping button becomes active when only one object is selected that has a texture applied. In that situation, clicking on the button opens the following dialog.



Again, if the Change Mapping dialog is not available, it may be for one or more of the following reasons:

- **No object is selected.**
- **More than one object is selected.**
- **The object selected does not have a texture applied.**
- **The selection is an instance of a shape and the texture was applied on a different level, thus it is not controlled at the current level.**
- **The object selected has a texture applied (at this level), but does not require mapping control.**

NOTE: *A distinction must be made between a texture using global surface property settings ONLY and a texture that contains surface maps. If there are no surface maps in the surface texture, it does not matter from what direction the texture is applied, or the scale, since*

its properties are the same across the entire surface. Thus, the Change Mapping button is not active when an object using that type of texture is selected. If the texture has at least one surface map, then the Change Mapping button becomes active so you can control the position and scale of the map(s).

As mentioned earlier, solid textures also have mapping considerations, but do not require all of the controls in the Change Mapping dialog. This is because mapping type and tiling style are not an issue with volumetric / procedural maps. However, scale and position are important, so the Change Mapping button is available for limited functions on solid textures.

A full description of the Change Mapping dialog — including a discussion of each control within the dialog — is presented later in this chapter

Previewing the work as you go

Preview renderings are an integral part of the texture application process. During the fine-tuning phase, you will probably want to take several "snapshots" to check the appearance of the surface maps on an object. Attention to mapping details during assembly of the model can eliminate rendering surprises later when you commit to the final image. It is very frustrating to find an image didn't turn out right when you begin the final rendering, then return — after a few minutes, the next morning, or a couple of days — only to find that you must make adjustments and start over. While rendering preview images throughout the project takes some additional time, it can actually help the project go faster.

Of course, you will want these preview snapshots to render as quickly as possible. You will want to turn off all non-essential rendering effects, disable anti-aliasing, and perhaps even lower the sampling rate for surface maps. This can be done manually or you can use the Draft and Good settings from one of the sub-menu choices in the Tool Palette's pull-down menu (at the bottom). These are pre-defined rendering settings that control all the variables automatically. Rendering effects and options are discussed in detail in *Chapter 8: Rendering.*

How to make maps to correct scale and resolution

Ever-present memory limitations dictate that surface maps be kept as small as possible. Yet, they must also be large enough not to

appear pixelated when sized to fit the target object. But how big is big enough? What other factors must be considered when deciding how large or what resolution is required to maintain the surface maps' quality when rendered?

In general, pixelization of surface maps can be avoided by keeping their scale at 100% or less. Yet surface maps can sometimes be too small to completely cover an object. Surface maps scaled larger than 100% can often render with good results if the object appears smaller than actual size in the rendered image, but this is not usually recommended. In those situations, if possible, you should try to use the image map so that it repeats or tiles the map across an object's surface rather than scaling it larger.

These are all considerations that must be planned ahead of time when constructing a surface map. Any number of full-color paint-type applications are available to aid in this process. The important thing to remember is not to over-do the resolution factor. Use what you need, but be aware that you have a finite amount of memory with which to model, create / load / apply textures, and render.

An example of calculating a map's resolution requirements

There are several methods to calculate the resolution requirements of surface maps. Once you understand the concept, you may have a preference that is easier for you to visualize. The one presented here is for concept only; real projects are rarely this clean, mathematically speaking.

Suppose you are modeling a room in which a picture is hanging on the far wall. Except for the picture, all of the other textures are either simple colors or solid textures. The surface map to be used for the picture MUST be of sufficient resolution to render clean — without pixelization or "jaggies."

This is the focus of the example: How do you calculate the minimum resolution required of the image that is to be applied as a surface map to the picture?

Further suppose that the final image is to be rendered at 200 d.p.i. This is determined because you plan to use it as part of a flyer that will be printed with a 100 line screen. Thus, 200 d.p.i. is the least resolution that you need for the rendered image to print with any expectation of good reproduction quality.

About image resolution

Why would a 100-line halftone screen dictate using a 200 d.p.i. resolution for the rendered image? For background, there are four types of resolution that are important to a digital image:

- **Color depth.** This is the measure of the number of bits of information per pixel. This resolution determines how much color information is stored in the pixels within the image. The common options for this type of resolution range from 1 bit per pixel (B&W) to 32 bits per pixel (16.7-million colors)
- **Dots per inch.** This is the measure of the amount of information in an image. The image resolution and its dimensions determine the overall size of the file. The greater the size, the more memory it requires and the more disk space it requires to store.
- **Halftone screen.** This is the measure of the density of the halftone ruling or frequency. Because the halftoning is arranged in lines, this resolution is measured in the number of lines per inch.
- **Device resolution.** This is the measurement of the number of dots per inch that the output device can produce (such as a laser printer at 300 d.p.i. and an Imagesetter at up to 3300 d.p.i.). The higher a device's resolution, the more levels of gray it supports. The maximum levels of gray is 256.

It does not matter whether the image is scanned, captured from a video source, or rendered in Vision 3d, every image you plan to output must be concerned with these resolution parameters. Of

these, the color depth and dots per inch (image resolution) are the ones you must be concerned with here.

Your image's resolution is a critical factor in determining the quality of the printed output. If the resolution is too low, the PostScript language in the output device will use a single pixel's color to create more than one halftone dot. This results in pixelization, or a very coarse-looking output. If the resolution is too high, you will send more information than the output device needs, taking longer to print than it should.

Determining the best resolution for the output

The best resolution for the printed output is dependent upon the quality of the output you want as well as on the resolution of your printer. If you are printing to a high resolution device, you will need a higher image resolution than if you are printing to a low-resolution one. Low resolution devices support fewer levels of gray for a given halftone screen density. That means fewer pixels are required to supply sufficient data to create the halftone dots, so you don't need as high a resolution.

There's one more point. Just because your trusty, rusty laser printer has 300 d.p.i. device resolution, that does not mean you can print to it with a 300-line halftone screen. In fact, the recommended halftone density for a 300 d.p.i. laser printer is only 53 l.p.i. And that only gives you 33 levels of gray for tonal gradations. Pretty wimpy, huh?

Digital halftone screen "spots" are made up of printer dots. The number of dots in each spot corresponds to the number of gray levels the spot can represent. The larger the spots, the more dots there are to represent the levels of gray, at the cost of a coarser screen. Finer (or denser) halftone screens create smaller spots composed of fewer dots. This may sharpen details, but reduces the number of grays that can be represented, resulting in severe posterization.

Here is the formula for calculating the relationship between gray levels, screen frequency and resolution:

$$\left(\frac{\text{Output Resolution}}{\text{Screen Frequency}}\right)^2 + 1 = \text{Available gray levels}$$

Imagine you are printing to a 300 d.p.i. laser printer, using a 60-line screen. That means there are 60 halftone spots per inch. Since

there are 300 dots per inch and each halftone spot is 1/60th of an inch, each spot contains 25 laser dots (in a 5 x 5 square).

Choosing the optimal screen frequency

Output Device Resolution *(dots/inch)*	Screen Frequency *(lines/inch)*	Available levels of gray
300	53	33
300	75	17
300	100	10
600	53	129
600	75	65
600	100	37
1000	53	356
1000	75	179
1000	100	101
1200	65	341
1200	85	200
1200	100	145
2400	85	790
2400	100	577
2400	150	257

Conclusion: On a 300 d.p.i. laser printer and printing with a 60-line screen, there are only 26 possible levels of grays. Having ONLY 26 levels of gray to work with results in noticeable posterization.

Okay, so now that you know the need for halftone screens and how to figure the levels of gray available, you can decide on the halftone density. But how does that relate to image resolution? That is the simplest part.

AS A GENERAL RULE, to produce sufficient quality for a given halftone density, the image's dots-per-inch resolution should be two to two-and-one-half times the halftone screen density used. That's it! It's that simple, but you must know the halftone density first.

BACK TO THE PICTURE EXAMPLE

In this example, your first thought might be that the surface map for the picture must also be 200 d.p.i., but that is not necessarily the case. If the picture in the model to be rendered appears at actual size then, yes, the image for the map would have to be 200 d.p.i. But, if the apparent size of the picture is one-half its actual size in the rendered image, then you would only need a map of 100 d.p.i. If the picture is one-quarter actual size, the map would only need to be 50 d.p.i. Do you see the pattern?

Now look at it the other way. If the image you are going to use is 72 d.p.i., and the amount it is being reduced when rendered is known, what is its effective resolution? Just as reducing or enlarging an image in a page layout program increases or decreases that image's effective resolution, the final resolution for the map is based on its apparent size in the final rendered image.

Note that none of this discussion is concerned with the scale at which the image is mapped to the object. That is a separate issue. It is the final rendered size compared to the original size that determines the resolution requirement.

In this example, the original image resolution was 72 d.p.i, but its apparent size in the rendered image is just under one-third of the actual size (.32 to be exact). The formula to calculate the effective resolution is as follows:

ER = 72 dpi / .32 = 255 dpi

To refresh your memory, when dealing with fractions, you convert to decimal format and divide. (You do know that .32 is the same as 32/100, don't you?) This calculates out to 225 d.p.i. as the effective resolution of the picture…more than enough for the 200 d.p.i. needed for the rendering. Obviously, using a higher resolution map for this project would have been excessive and taken more memory than necessary.

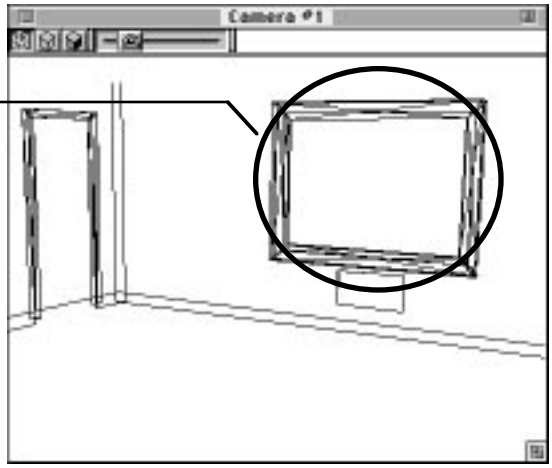
Of course, things get more demanding if the rendered image is going to be larger than actual size, such as for use as a poster. If a surface map will be rendered at 1.5X its actual size, then the resolution of that map must be 1.5X greater than the resolution of the rendered image.

Original image

Dimensions:
544 pixels wide by
333 high (actual size)

Rendering image

Dimensions of picture in image: 177 pixels wide by 136 high (apparent size, with some cropping of the original's width)



Effective resolution

The rendered image is .32 of original size. Thus, the apparent resolution of the original image in the rendered image is 225 dots per inch

Whether larger or smaller, the question is the same: *WHAT IS THE SURFACE MAP'S ACTUAL SIZE COMPARED TO ITS APPARENT SIZE IN THE FINAL RENDERED IMAGE?*

Map resolution in animations

Determining the resolution requirements of a surface map is relatively straight forward for still images. The size of the map in the finished image will be fixed, so you can work backward to figure the map's minimum requirements. However, in the case of animations, those requirements can get rather more difficult to calculate as the object moves from background to foreground during the animation.

The general rule here is to determine the largest size each object in an animation will appear in the rendered frames, then use that as the basis for establishing the minimum resolution of the maps that may be used on the object. Yes, it does take some time to lock down these additional details, but the results are worth it if you can avoid a map becoming pixelized in that critical moment it moves close to the view plane at larger-than-normal size.

On the other hand, the eye is more forgiving of detail when an object is moving than when it is still, so you can often get away with lesser resolution than the formulas would suggest. Again, you should plan the surface mapping to factor in sufficient resolution, but avoid resolution overkill that gobbles up memory you may need for something else.

Re-sampling the pixels in surface maps

Under System 7.x, PICT images with resolutions greater than 72 d.p.i. may be used as surface maps without stripping away the "excess" data. When an image with higher than 72 d.p.i. resolution is loaded, the resulting map is re-sampled to 72 d.p.i. and made correspondingly larger, rather than purging the extra data. Thus, a 3 inch by 3 inch image at 144 d.p.i. (with a size of 547K when opened in Photoshop) becomes a 6 inch by 6 inch map at 72 d.p.i. upon loading into Vision 3d. It contains the same number of pixels and uses the same amount of memory in both cases.

The size of such a higher-resolution map must be adjusted to fit the object it was intended for. Thus, if you have a rectangle that is 3" by 3" square, and the 3" by 3" / 144 d.p.i. map becomes 6" by 6" / 72 d.p.i. upon loading into Vision 3d, you know that the scale setting in the Change Mapping dialog must be 50% to make it fit the 3" by 3" object. And the map's effective resolution is again 144 d.p.i. (If in the rendered image, the 3" by 3" object actually appears only 2" by 2", then the effective resolution of the map becomes 216 d.p.i.)

Alternately, you could just switch the Change Mapping dialog to “coverage” mode and simply set the values to 100%. That automatically reduces the map’s actual size to fit the object. The choice of modes — coverage or scale — depends on how you want to visualize the mapping process

HOW BIG CAN A MAP BE?

Loading a higher-resolution image for a surface map often results in a very large map that must then be scaled smaller to fit the designated object, thereby obtaining the desired increase in effective resolution. The rendered image often looks much sharper because of the map’s tighter pixels.

This image exceeds the recommended storage requirements of 256 KBytes. It should be reduced to a smaller size.
Keep Large
Reduce

While some projects require higher-resolution maps to improve image quality in the finished project, high resolution images also require a greater amount of memory. This is memory you may not have or need to use in some other way. Experience indicates that the vast majority of maps can be less than 256K and perform their function quite well in a “normal” project.

To that end, when you load a map into Vision 3d that is larger than 256K (through any combination of size and color depth), you will see an alert asking if you wish to reduce the map or keep it full size. Note that this limit is advisory and you may keep the image full size, if needed. The alert is presented so you are aware of the need to be conservative with the program’s memory requirements. If you choose to reduce the map, Vision 3d maintains the aspect ratio during reduction and will shrink it just enough to come under the 256K limit.

Alternately, you may find that reducing the color depth (from 32-bit to 8-bit, for example) can sufficiently lower the memory requirement without shrinking the dimensions, and the image will take one-fourth the amount of memory.

The size requirements of a map depend on many variables, i.e. — scale of the object, scale of the map, tiling style, details in the map, actual size of the final rendered image, etc. The optimal situation is to use a map slightly bigger than required, then apply a small scale reduction through the Change Mapping dialog box in the Textures menu. This helps insure no pixelization.

Generally, you should avoid sizing a map larger than 100.0 in the Scale mode of the Change Mapping dialog once it has been applied to an object, since that can induce pixelization of the map, even when its apparent size in the final image is smaller. A scale of 100.0 means 100% of actual size. If you find you are always having to scale up maps, you should consider increasing their size…but do it sparingly.

NOTE: *There is a maximum size limitation imposed on Vision 3d by the operating system, via the QuickDraw imaging model used in every Macintosh. This applies to images used for surface maps, as well as the rendered images. At this writing, the limit is 4,000 pixels on the horizontal dimension and 16,000 pixels vertically.*

How much memory will it take?

If you plan to insert any textures from a commercial library, or create your own textures containing surface maps, use the following formulas to determine the memory requirements of each PICT file you include in a texture:

- **Color maps.** Multiply the height (in pixels) by the width (in pixels), then multiply that number by four. The result is the number of bytes required to store a 32-bit image as a color map in memory. Color maps stored in the model in 8-bit format are one-fourth the 32-bit version.
- **Grayscale maps.** Multiply only the height by the width (in pixels). The result is the number of bytes required to store the image as a gray-scale map. Grayscale maps are stored as 8-bit images.

Quick Tutorials

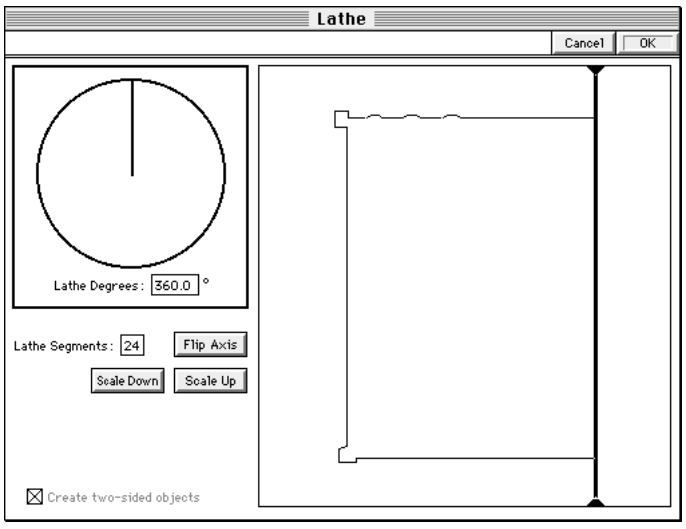
The object of these quick tutorials are to give you a basic understanding of how textures work in Vision 3d. You will learn how to load a texture, how to create a texture, how to apply textures, and how surface maps work. You'll also learn how to

compose a texture with multiple surface properties using composite maps.

Exercise 1: Label on a can

In Exercise 1, you will learn how to use a PICT image to make a simple surface texture (a label) and how to apply that texture to a cylindrical object in a Vision 3d model. You'll also learn how to adjust both the size of the object and the mapping parameters to make sure the texture fits correctly on the rendered object.

1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Lathe a can-object.** Choose the polyline tool as you have used in previous tutorial exercises of other chapters. Using the same techniques as in these previous exercises, draw an outline similar to the one shown in the illustration below.



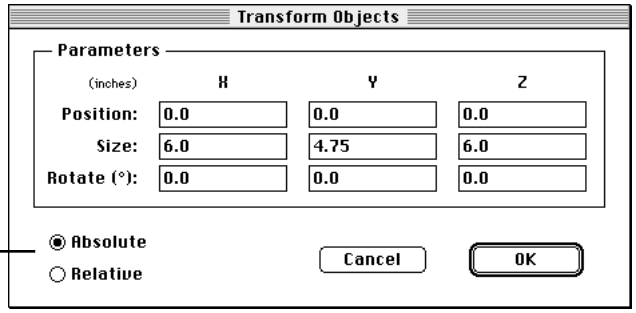
Don't worry about getting the size right. You can adjust that later. Once the outline is drawn, choose the Lathe modeler. Match the axis position and settings shown, then click on the OK button. Vision 3d draws the can in the model window.

3. **Adjust the size of the object.** With the 3-D can object selected, pull down the modeling menu and choose the Transform command. Vision 3d opens the Transform Objects dialog box.

   Make sure you are working in Absolute mode, by clicking on the Absolute radio button in the lower left of the dialog. Match the numerical values in each field with those in the following illustration and click on the OK button. Vision 3d redraws the can in the model window using the precise values you set in the Transform Objects dialog window.

Set the Position numbers to zero (0), Size to 4, 4.75, 4, and rotation to zero (0).

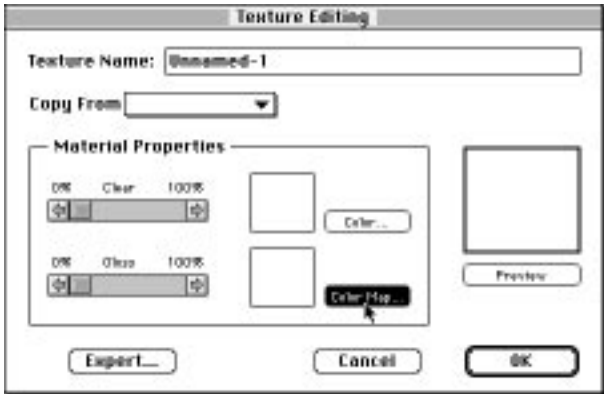
Make sure that Absolute is selected



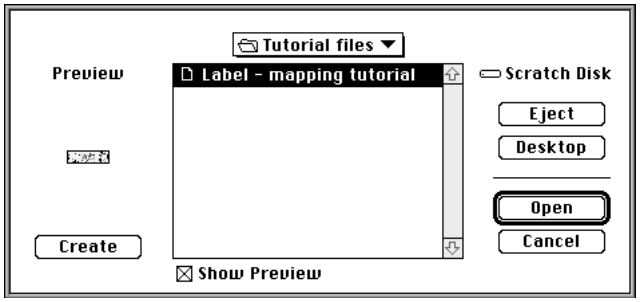
The Transform command gives you another method to scale, position, and rotate objects on all three axes. You already know you can perform these operations interactively using the click-and-drag method with the tools on the Tool palette. You have done this in earlier tutorial exercises.

Or you can use the Transform command and input numerical values for more precise control of object position, size, and rotation. In this exercise, you need to know the precise dimensions of your can object so you can ensure the label's proportions will be correct and the texture will wrap around it in a realistic way when the object is rendered.

4. **Create a new texture to place on the can.** On the Textures palette, click on the New button. Vision 3d opens the Texture Editing dialog for a surface-type texture.

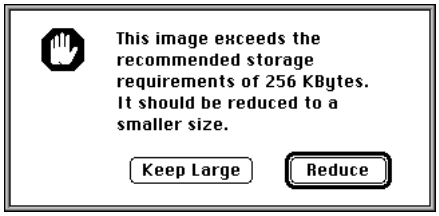


5. **Click on the Color Map button inside the Material Properties box.** Vision 3d opens the Picture dialog box.
6. **Click on the Get Picture button within the Picture dialog.** This opens the Load File directory dialog, which allows you to navigate to the location of the PICT file you want to use as a texture.



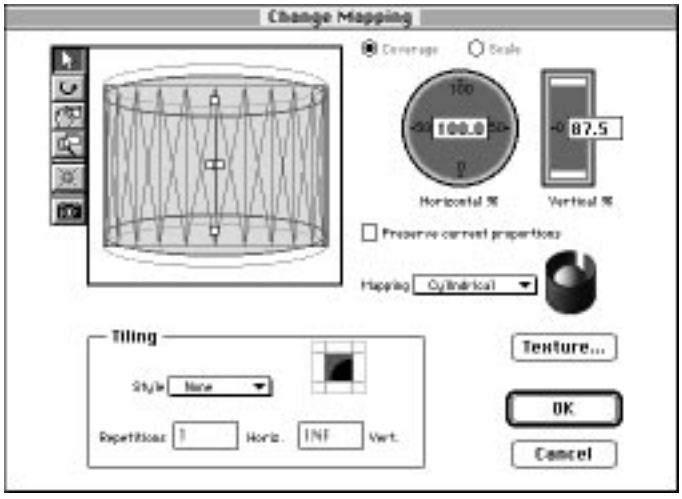
7. **Find the "Label - mapping tutorial" in the Tutorial files folder.** Choose this file and click on the OK button. Vision 3d opens the PICT in the Picture dialog window.

8. **Switch the Change Depth field to 8-bit and click on the OK button.** This accepts the surface map as an 8-bit color image. The surface map that you have loaded was saved in 8-bit color originally. But remember, all color maps are loaded into the model in 32-bit color by default. Click on the pop-up menu to change the color depth of the surface from 32-bit to 8-bit color, then click on the OK button. Loading a surface map as 8-bit color requires one-quarter the amount of memory as a 24-bit color image.
9. **Click on the Keep Large button.** Since the file size of the image is greater than 256K, Vision 3d notifies you the color map is larger than the recommended size limit. You are given the option to reduce the dimensions of the color map or keep it large – that is, at its original dimensions.


This image exceeds the recommended storage requirements of 256 KBytes. It should be reduced to a smaller size.
Keep Large
Reduce

This prompt appears only when the surface map file being loaded exceeds the recommended color map file size limit of 256K. Because this map was prepared so it would fit on the can at original size, you want to keep it full size. The surface map is placed into the color map field in the Texture Editing dialog window at the size it was created.

10. **Name the texture.** Type “Can Label” in the Texture Name field of the Texture Editing dialog. Then click on the OK button. Vision 3d returns you to the modeling window and renders a preview of the texture on a tile in the Textures palette.
11. **Make sure the Can Label texture is applied to the can-object.** If an object in the model window is selected when you create a texture, the texture is automatically applied to that object.
12. **Adjust the texture to fit the can object, then click on the OK button.** Click on the Change Mapping button on the Button bar. Vision 3d opens the Change Mapping dialog window. In the Mapping Preview box, you see a representation of the texture overlaid on a representation of the can object.



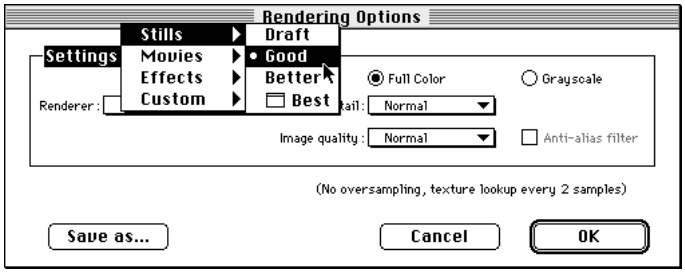
Set Tiling Style to None, and Mapping to Cylindrical, as shown. Make sure the Preserve Current Proportions checkbox is NOT checked, then match the Horizontal and Vertical percentages

shown here. If necessary, center the texture overlay so it aligns with the object representation in the Mapping Preview box, as shown. This can be done easiest by double-clicking on the pointer tool.

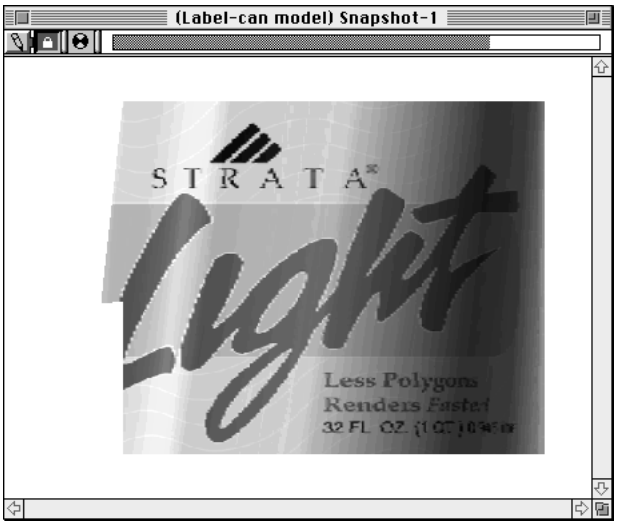
Clicking on the OK button puts you back in the model window.

13. **Set rendering quality.** To see whether our label texture has been placed correctly on the can object, you must render the model. The first step is to set the quality of the rendering. This can be done with the pull-down menu on the Tools palette. Or you can use a different method through the Rendering Options command in the Rendering menu.

    Vision 3d opens the Rendering Options dialog box. Pull down the Settings menu and choose “Stills.” Next, choose “Good” from the sub-menu. Clicking on the OK button returns you to the model window.



14. **Render the model.** Choose the large camera icon from the bottom of the Tool Palette. Notice when you move the cursor into the model window it changes to a camera, indicating the rendering tool is active. With the camera cursor in the model window, click once. Vision 3d begins to render everything in the model window.



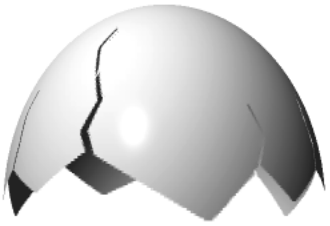
Your finished rendering should look similar to the one shown in the following illustration. You can try rendering different variations of this model. For example, you might add perspective by changing the three-position lens control to Normal, or change the view angle by choosing another one from the View pull-down menu.

In this example, rendering different views will reveal previously unseen areas of the label. You may also want to experiment with other rendering settings in the Rendering Options dialog.

Rendering is discussed in detail in *Chapter 8: Rendering*.

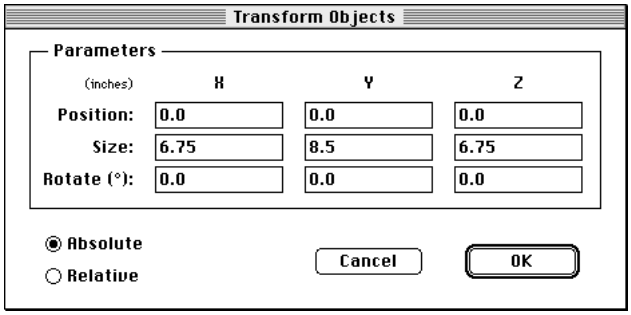
Exercise 2: Cracked eggshell

In this exercise, you'll learn how to create a more complex texture. You will place several simple PICT files (like those created by a paint program, for example) into Vision's expert texture mapping fields. These maps, in conjunction with the expert surface property fields, will form a texture resembling a cracked egg shell.



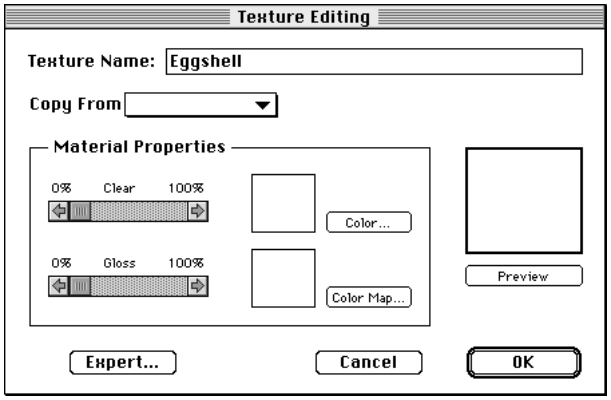
You'll apply this texture to a simple sphere primitive and render it, creating a finished image of a broken egg shell similar to the previous illustration. You'll also learn how to use a background environment as part of a rendering.

1. **Open a new model.** Make sure the three-position lens control is set to Orthographic, and the view orientation is set to Front.
2. **Draw a sphere in the model window.** Choose the sphere primitive from the Tool palette. Click once in the model window. Vision 3d draws a sphere in the model window.
3. **Re-size the sphere.** Match the values shown in the following illustration of the Transform dialog box. Basically, you are stretching the sphere vertically so that it becomes more egg-like in appearance.

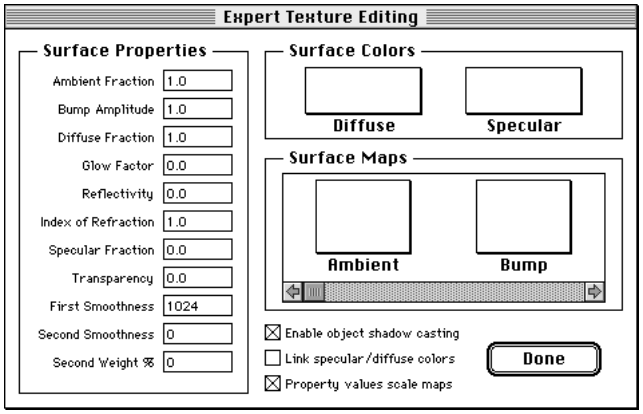


If the re-scaled sphere is too large to fit in the model window, use the "Fit Views to All" command in the Windows menu. You may also want to look at the sphere from the Top view or a side view to verify the object's geometry. If you do change the view, be sure to return to the Front view before proceeding.

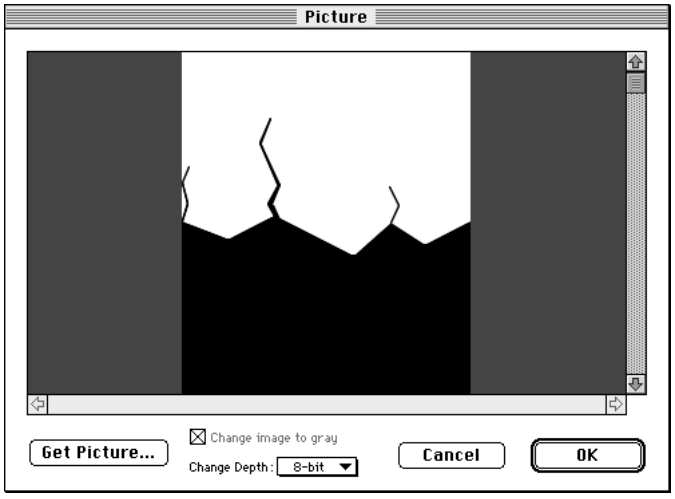
4. **Create a new texture.** As before, use the New Texture button in the Textures palette. The texture editing dialog box for creating a new surface texture opens. The first thing to do is to give the new texture the name of "Eggshell."



5. **Click on the Expert button.** This is found in the lower left of the Texture Editing dialog. Vision 3d displays the Expert-level Texture Editing dialog box.

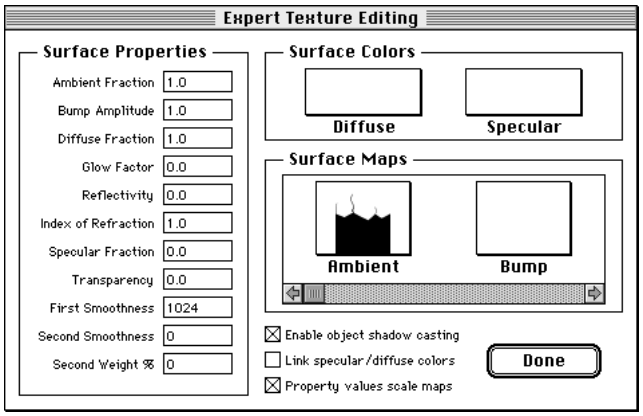


6. **Click once inside the Ambient field within the Surface Maps area of this dialog.** Vision 3d responds by displaying the Picture dialog window. Through this dialog you are going to get the PICT image that will be placed in the Ambient surface mapping field.
7. **Click on the Get Picture button.** The standard file directory dialog opens. This dialog allows you to find and open the PICT file you want.
8. **Load the "Eggshell - amb/diff/spec" file.** This is in the Tutorial folder within the StrataVision 3d application folder. This is a grayscale PICT.



Vision 3d loads the PICT and inserts it into the Eggshell texture as the Ambient surface map.

9. **Click on the OK button.** Vision 3d returns you to the Expert Texture Editing dialog. You can see that the Eggshell - amb / diff / spec PICT has been placed in the Ambient field.

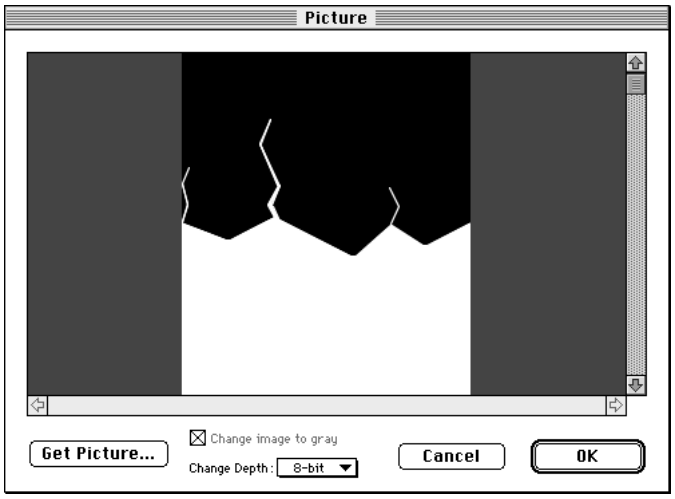


10. **Load the Diffuse surface map into its field.** In the Surface Maps section of the Expert dialog, scroll to the right until you see the Diffuse field. Then, click on the Diffuse rectangle to insert the same “Eggshell - amb / diff / spec” image in this field. Follow the same steps used to place the map in the Ambient field.
11. **Copy the map to the clipboard.** Before clicking on the OK button of the Picture dialog to insert the map into the Diffuse map field, use the Copy command in the Edit menu to copy the image in this dialog to the clipboard. This is done to demonstrate an alternative method for inserting the same map in several fields.
12. **Load the Specular map field with the same map.** Of course, you could load the map into this field just as you did the other two fields…by now this is old stuff to you. But, by copying the image to the clipboard, you won’t have to go to the hard-disk drive again.

    Scroll right in the Surface Maps section to find the Specular field (NOT to be confused with the Specular Color field). Click on the Specular field’s rectangle to open the Picture dialog as before. This time use the Paste command in the Edit menu to place the same image into this dialog, just as if you had loaded it from the hard disk.

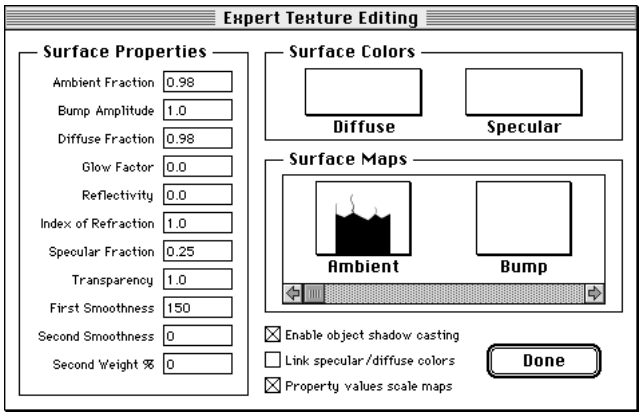
Of course, all the rules of the clipboard are in force here, just like any other application. That is, only one item is stored on the clipboard at a time and a copy or cut command replaces what was on the clipboard with the new item. So, to be able to paste the same map into multiple fields, you cannot use the copy command to temporarily store something else and still retain the map to be pasted into the additional texture fields.

13. **Load the "Eggshell - transparency" file into the Transparency field.** Use the same process as above. This is a different map, so it must be loaded from the hard-disk drive using the Get Picture directory dialog.



NOTE: *This map is inverted when compared to the first image used in the other fields. While all the fields work on the same principle — that of controlling the amount of a specific surface property with levels of gray in the desired pattern (as explained earlier in this chapter) — whereas the white area in the Ambient, Diffuse, and Specular maps is where you want light to reflect off the eggshell, the transparency map uses an inverted image because the white area in this map corresponds to that part of the sphere's surface you want to be transparent.*

14. **Set the Surface Properties fields to work with the Surface Maps fields.** Match the values in the surface properties to those shown in the following illustration.

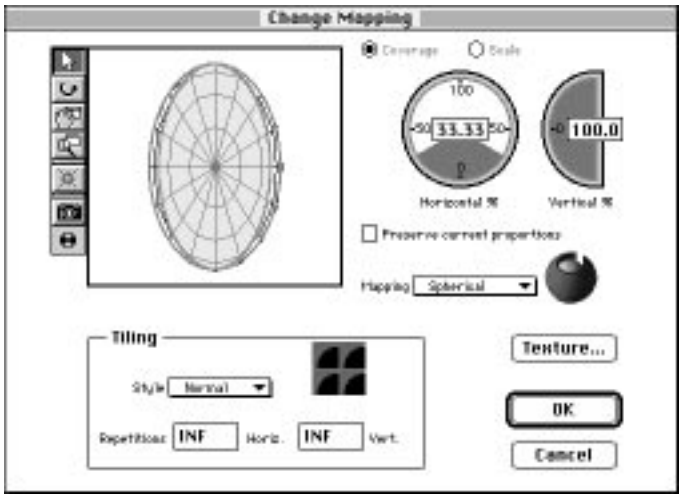


These create a slightly glossy surface with high ambient and diffuse light response (where the maps allow it), but also with transparency enabled (where its map allows that property).

15. **Return to the model window.** Click on the Done button in the expert dialog, then on the OK button in the primary texture editing dialog to return to the model window. Notice the Clear and Gloss slider settings in the Texture Editing dialog have changed to reflect the settings you entered in the Expert Texture Editing surface property fields.

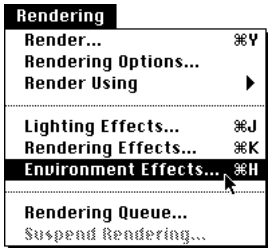
16. **Make sure the Eggshell texture is applied to the sphere primitive in the appropriate mapping style.** Use the Change Mapping dialog and the same procedure you used in the previous exercise.

    Set the mapping controls in the Change Mapping dialog to the values shown in the following illustration. These parameters will produce the appropriate appearance for the eggshell object when rendered.

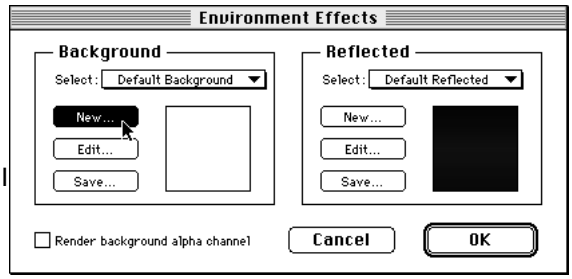


No explanation is offered for these settings here. This is a test! By now, you should be able to understand why this texture is mapped onto the object in this manner.

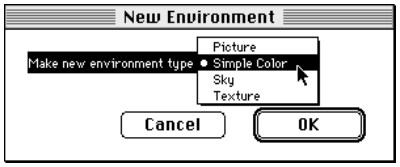
17. **Add a background environment to the model.** Pull down the Rendering menu and choose the Environment Effects command. Vision 3d opens the Environment Effects dialog window.



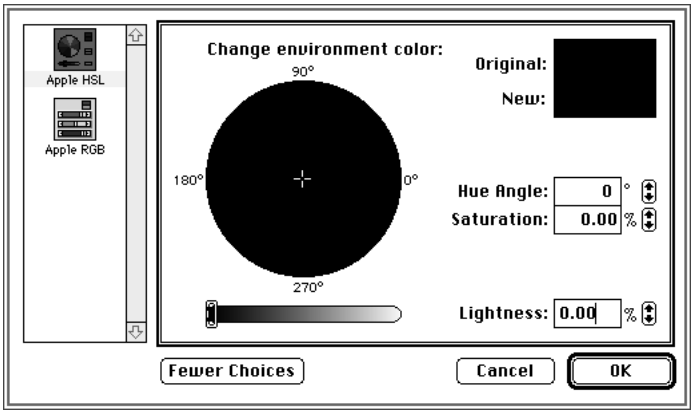
18. **Create a new background.** On the left side of the Environment dialog box, click on the New button. Vision 3d opens the New Environment dialog window.



19. **Choose Simple Color as the new environment type.** Set the Make New Environment Type pop-up menu to “Simple Color.”



20. **Click on the OK button.** Vision 3d brings up the Macintosh color picker.



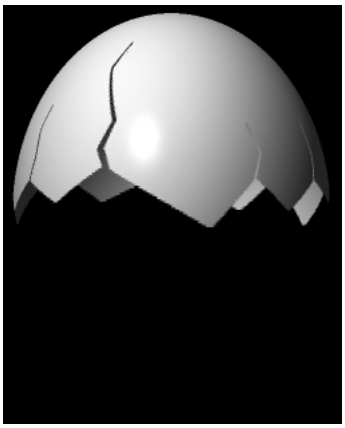
If the Color Picker dialog that appears on your screen does not look exactly like the one shown in the above illustration, that’s

okay. This dialog is part of the ColorSync extension. The default color picker works just fine.

21. **Set the color to black and click on the OK button.** Vision 3d returns you to the Environment Effects dialog. Notice the background environment preview box on the left side of the dialog is now black, showing the background environment color you chose in the color picker.
22. **Click on the OK button in the Environment Effects dialog.** Vision 3d returns you to the model window.
23. **Render the sphere.** In Exercise 1, you used the Rendering Options command in the Rendering menu to set the rendering parameters. This time, click on the pull-down menu at the bottom of the Tool palette.

Choose “Stills - Better” as the rendering parameters from the sub-menu, then release the mouse button. The Camera tool automatically selects. Move the cursor to the model window and click once to initiate a rendering.

The “Stills - Better” rendering setting will use the raytracer to render the sphere model. You will see successive passes of the renderer as it refines the image. This continues until the rendering is finished.



Your finished rendering should look similar to the one shown here.

In this exercise, you discovered how surface maps placed in the expert mapping fields work together with the surface properties values to create surface detail (or in this case, the absence of surface through a transparency map) on an object in your model. This should help you better understand the power of surface maps to alter the appearance, and even the apparent geometry, of an object or portion of an object.

Textures on light sources

Lighting in a computer model in many ways is analogous to lighting for a theatrical play, i.e. — the scene is assembled, then lit for dramatic effect. A little drama in a rendered image can significantly improve its impact.

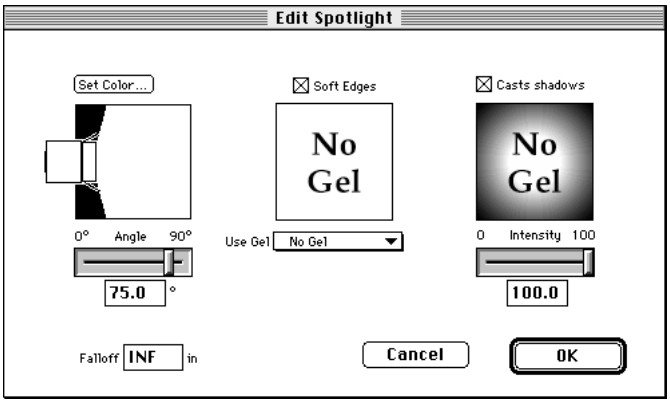
A light is a source of illumination. Light sources interact with surfaces (and the textures that define surface properties) to determine the amount and type of light that a surface reflects, as well as cast shadows on other surfaces.

The use of lights is discussed in *Chapter 6: Lighting*. Lighting is mentioned here as it relates to the use of textures on light sources.

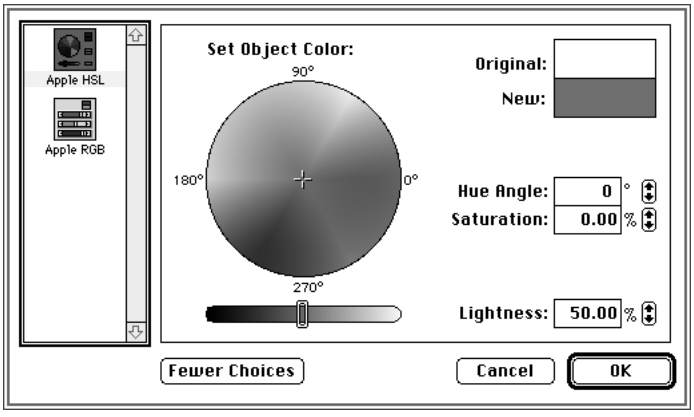
Giving lights color…and more

All lights have color. The default color for light sources is white. A light source's color — whether it be a directional, spot, or point light — may be changed by opening the Edit Light dialog. For directional light sources, this is done by selecting the light source and clicking on the Edit button found on the Lighting palette.

For local light sources, you can select the light source's icon and choose the Edit Primitive command in the Modeling menu. Alternately, you can simply double-click on the light source. The following illustration showing the spot light's edit dialog is representative of the other light sources as well.



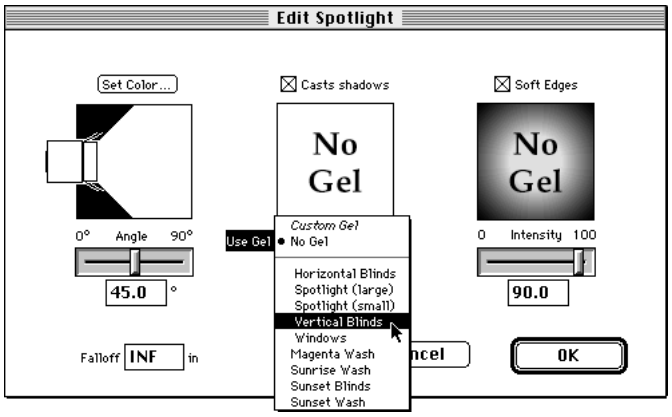
Lights can be any color. It is a simple matter of specifying a color in this dialog box for the light source. Clicking on the Set Color button opens the Apple Color Picker available in all applications.



Of course, there are other controls in this dialog for intensity, falloff, soft edges, and more. These are explained in Chapter 6. The focus here is in applying textures to light sources.

USING PRE-BUILT GELS

Within the light editing dialog, you can also apply specialized "Gels" to a light source. This is done by clicking on the Use Gel pull-down menu. The gels listed in that menu are from the Gels folder in the StrataVision 3d application folder.



Gels are a type of texture, specifically optimized for use with light sources. When a gel is selected, it is automatically loaded and applied to the light source with the proper orientation and scale so that it works correctly in most situations.

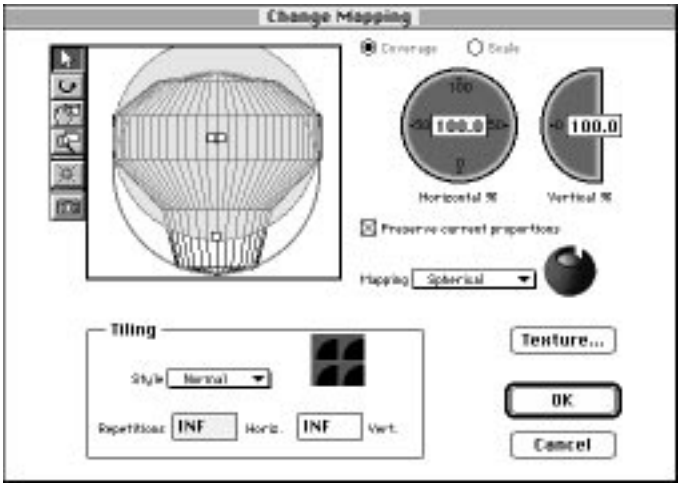
MAKING YOUR OWN GELS

Gels are the preferred method of using light sources as projectors of images. As noted, several pre-built gels are included in the Vision 3d package. If none of the gels provided fit your needs, you can create your own gels by selecting "Custom Gel" in the Use Gel pull-down menu.

The familiar surface texture dialog box appears in which you can load any combination of surface maps you wish. While you have access to all the surface texture editing controls, some of the settings are not relevant to use on a light source. For example, since the

texture is being projected instead of mapped onto a surface, if you set a high gloss or reflectivity value it will just be ignored.

Once the gel / texture is defined, click on the OK button to return to the Light Edit dialog. The texture is mapped to the light source using a planar mapping style at 100% coverage of the icon, but you may change the mapping parameters by clicking on the Mapping button.



The familiar Change Mapping dialog appears in which you can adjust the mapping parameters to the specific needs of your custom gel. You may re-edit your gel texture and its mapping parameters as often as needed. It is not uncommon to require a couple of tries to get a custom gel just right.

Using regular textures as a gel

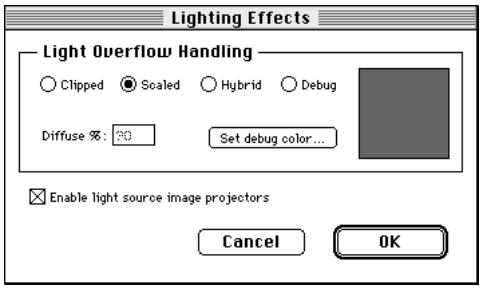
Of course, you can also apply “non-gel,” standard textures manually to local light sources, just like any other object. You then set the mapping type, scale, and tiling method through the Change Mapping command directly.

Spot light and Point light sources can accept information to define the nature of their light either way. You can apply this information in the model window with textures or in the light

editing dialog with gels. However, the relevant characteristics a texture or gel can impart to a light source include only color and the projection of color surface maps. Other settings like transparency, reflectivity, bump map, etc. (if included in the texture) are ignored for light sources. If the texture has a color defined, the illumination coming from the light source takes on that color. If the texture includes a color surface map, the light source can project the color map as a component of its illumination effect.

Whenever a surface map is used in a texture applied to a light source, that light source acts as a projector. Mapping type, scale and tiling parameters in the Change Mapping dialog are available to adjust the texture for that light source to obtain the projection desired. All mapping types are applicable with varying effects, i.e. — planar, spherical, cylindrical, etc.

Remember, as with objects, surface maps applied to light sources are direction-sensitive. That is, the orientation of the texture as it is projected from the light source is determined by the active view from which it was applied.



One more point, the “Enable Projectors” field (in the Lighting Options dialog) must be turned on for surface maps to work on light sources. The default is for this field to be turned on. If you have problems using this feature, you should check the “Enable Projectors” field first, just in case this option has been unknowingly turned off.

Directional Light Sources

The Lighting palette displays the directional light sources active in the current model. These light sources are sometimes also referred to as global lights. Directional lights are not the same as local light

sources or geometric objects. Their settings can only be edited through the Light Edit dialog and textures can only be applied as a custom gel via the dialog controls.

Spot and Point Light Sources

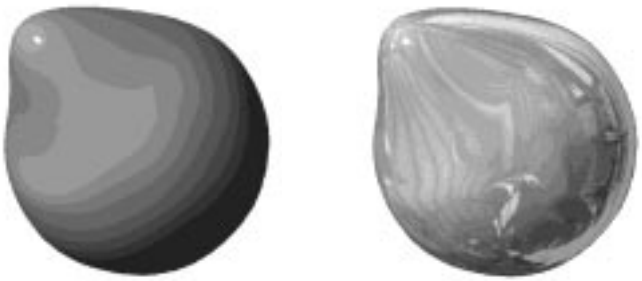
These light sources are inserted into the model as object "icons." That is, they are three dimensional in nature, but the scale of the icon has no affect on the size and intensity of the light source.

A point light source is omni-directional. A texture containing surface maps may be applied from any direction. Point lights would typically use a spherical mapping when applying your own textures or custom gels.

Spot lights are directional in their illumination, like the global lighting, but local in their effect, like the point light source. Typically, textures using surface maps are applied from the front or back and set for Planar mapping. Applying the surface from the back projects the image as it is loaded into the texture. Applying the surface from the front reverses the image when projected.

Textures in environments

Vision's environment feature allows you to apply various images or patterns to an imaginary sphere surrounding the entire model. You can designate that the environment appears in the background or only reflects off the shiny objects in your model. Environments can add to the realism of a model by implying more detail than is actually modeled. An added benefit is that reflective environments work with the Phong renderer.



Both images were rendered using Phong shading. Notice that for the right hand image the "Room Reflected" environment was loaded.

One of the environment types supported allows you to use textures for the environment's image or color. Typically a texture - based environment includes a color surface map of some kind. Of course, you can use a texture for simple color in an environment if you wish, but that would be inefficient since solid colors can be implemented using the Simple Color environment type

Texture-based environments may use either surface or solid textures. Several texture-based environments are shipped with the product. You may use the ones provided, or create your own. When creating your own environments, you can use one of the textures provided or make a new one from scratch.

Applying a texture to the environment sphere is not the same as applying it to an object or light source in the model. It is important that you realize that environment textures have special considerations concerning size and tiling repetitions. The mapping style is always spherical; there is no option.

Obviously, every situation is different when making your own environments. To aid in understanding how to efficiently use this feature, you may wish to inspect one or more existing environments. See how large to make the map(s) and how to set the scaling values to best fit the environment sphere, so they appear as expected in the rendered image.

Additionally, once you have created an environment that you wish to reuse, you can also save environment files for use with other models.

As you can see, you have total flexibility in the creation and use of environments. Textures are used for environments just like they would be used for objects, except that you cannot directly select the environment sphere. You must use the environment editing dialogs to apply a texture to the environment and scale it appropriately.

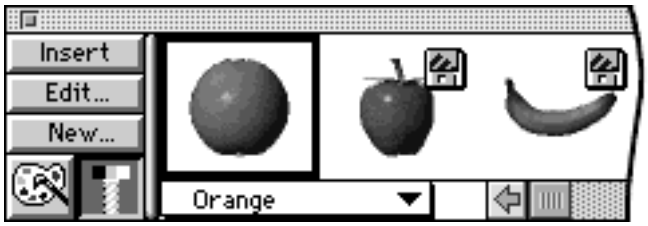
Note that textures used for environment purposes become part of the model, just as when applied to objects. They are saved along with the other textures in the model and will be in place upon re-opening the model for a later session.

Textures Palette

The Shapes and Textures palettes share the same space on the screen. Clicking on the Shape button in the bottom left of this palette

switches it to the Shapes palette; clicking on the Textures button switches it to the Textures palette.

Since so many of the issues involving shapes and textures are similar, you will find many aspects of Vision's interface share this parallel or dual-purpose functionality for shapes and textures.



Both Textures and Shapes use the same palette. The New and Edit buttons are reused while the Apply button changes to Insert.

The Textures palette contains a preview image and a listing in the pop-up menu for each texture available in the Textures folder. At the beginning of a session, the Textures folder is scanned and those are the textures displayed on the Textures palette. As you define new textures or load textures from other locations into the active model, they also appear on the Textures palette.

When a new texture is created in a model, it immediately appears on the Textures palette and can also have an image generated for the palette to help visually identify it. Texture files loaded from sources other than the Textures folder bring their own preview, which automatically appears on the Textures palette when loaded.

In addition, files imported into a model (which automatically creates a new shape workspace to contain the imported objects) may contain textures. Any textures that come with an object that is merged into the active model are also given a space on the palette, even if they do not have a preview image for the palette. If a preview is not present, one is automatically created for a texture imported with an object.

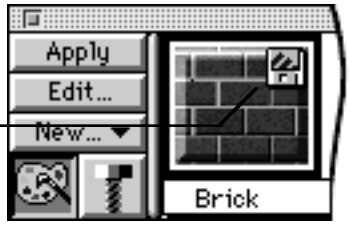
If more than one model is open, switching active models causes the Textures palette to redraw. This is necessary so that the palette can accurately show the currently-loaded textures for the active model.

Operation

Representations of all textures available to the current model are presented on the Textures palette. The available textures include those created in the model itself, those loaded into the model or those found in the Textures folder on the hard-disk drive upon startup of the application (but not loaded). A preview of these shapes may be viewed by scrolling their images horizontally in the palette window.

When scrolling left to right on the Textures palette, the textures created in or loaded in the active model always appear first in the palette's image preview area. Textures found in the Textures folder, but not loaded in the current model are identified by a floppy disk icon super-imposed on its preview image. Loaded shapes do not have the floppy disk icon visible.

Notice the disk icon on textures that are not loaded into the current document.



To select a texture on the palette, click once on its preview image. Only one texture can be selected at a time on the palette. A black rectangle appears around the selected preview. Alternately, you can use the pop-up menu to select a texture by name. Choosing a texture from the menu automatically scrolls the palette to show that texture's image and moves the selection rectangle to that texture.

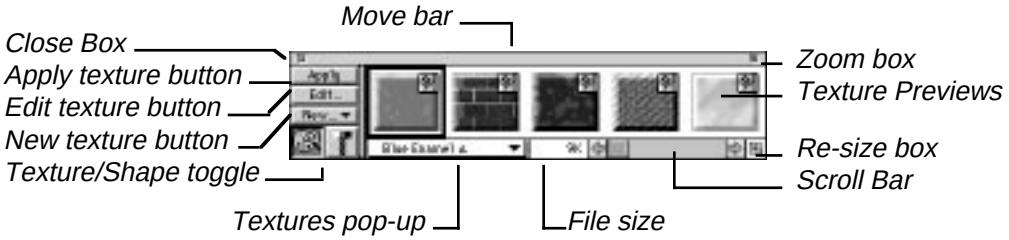
To load a texture from the palette, double-click on its preview image. The floppy disk icon is then removed from that image, indicating that it is now loaded in the model. Loaded textures appear at the front of the palette, followed by those available in the Textures folders but not loaded. Note that this process only loads the texture, but does not apply it to any objects in the model.

To apply a texture from the palette, select the object and the texture's preview on the palette, then click on the Apply button. Again, the active view determines the texture's application direction for mapping orientation.

The texture will be loaded into the model and applied to the object. As explained earlier in this chapter, the active view can be for the model space or a shape's workspace. Even if the texture is already loaded, clicking on the Apply button still performs the apply operation.

Controls on the palette

Controls found on the Textures palette allow you to manage the textures represented there. Note that if more than one model is open, as with other aspects of the interface, these controls have effect only in the active model.

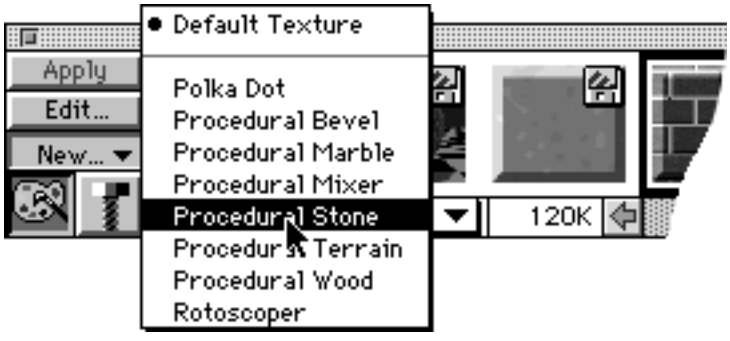


The Textures palette controls have been alluded to previously in this chapter in explanations of certain operations. They are presented here item-by-item for reference:

- **Move bar.** Use this control to position the palette on the desktop. Vision 3d remembers the position of the palette between sessions or upon showing the palette after it has been hidden.
- **Close box.** Click on this control to hide the Textures palette. This is the same as choosing the Hide Textures palette from the Palettes sub-menu in the Window menu, or clicking on the bottom right Hide/Show Palettes button on the Button bar.
- **Zoom box.** Click on this control to expand or contract the palette horizontally. Zooming out automatically expands the palette to the full width of the monitor on which it is displayed. Zoom in returns the palette's width to the size set by the re-size box. The palette does not expand vertically. If there are more shapes, they can be displayed on the palette at full width.
- **Re-size box.** Use this control to manually set the width of the palette on the screen. You can drag the re-size control horizontally. The widest setting possible is the full width of the desktop (even across multiple monitors, if available).

Note that the width you drag with this control "snaps to" the closest whole preview. You cannot set the palette's width to show only a partial preview image of a texture.

- **Scroll bar.** Use this control to scroll horizontally in the palette to view all the textures represented there. The scroll bar is active only when there are more textures than can be viewed at one time on the palette.
- **Textures preview area.** This area of the Textures palette contains a preview image for each texture available in the active model. Whether created in the model, loaded into the model, or still in the Textures folder on the hard-disk drive, each texture is given a place on the palette.
- **Apply button.** Click on this button to apply a selected texture to a selected object in the active model. The texture is loaded into the model, if needed, then applied to the object.
- **Edit button.** Click on this button to edit the texture selected on the Textures palette. A texture-editing dialog is opened for that texture. If the texture chosen to be edited is not currently in the model, it will be automatically loaded first. If the texture selected is already loaded, it is not re-loaded, but simply opens that texture's editing dialog box.
- **New button.** Click on this button to define a new texture. The new texture is created in the active model. By default, a surface-type texture editing dialog box is presented in which you may enter the name of the texture and define surface properties for that texture. Clicking on the OK button accepts the texture settings and creates that texture in the active model. Clicking on the Cancel button discards the texture definition and no texture is created.



If you hold down the mouse button while clicking on the New button, a pop-up menu appears. This menu contains a list of the texture extensions from which you can create a new solid-type texture. Highlighting one of the texture extensions before releasing the mouse button opens a texture-editing dialog box appropriate for that type of solid texture. Each texture extension provides its own texture editing dialog containing the fields required to define a texture based on that texture extension.

- **File size field.** This field indicates the size in Kbytes of the selected texture prior to loading. This is the size of the texture file as it is saved. The texture may actually take more space once loaded into the model, especially if any surface maps are used in that texture.
- **List of Textures (pop-up menu).** Use this menu to choose a texture on the palette by name. The menu lists any textures that are loaded in the active model first, with a divider between the loaded textures and those not loaded but available in the Textures folder. Within each group, the listing is alphabetical.
- **Loaded/On-disk indicator (floppy icon).** The presence of this icon indicates that a texture is not loaded in the active model. It is removed from the preview images of textures loaded in the model.

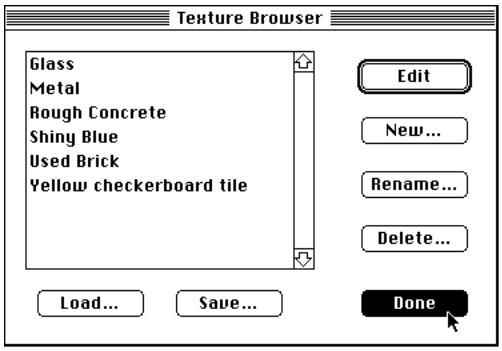
Commands and dialogs used with Textures

The following commands are considered pertinent to the handling of textures in the software. A detailed explanation is included under each command in the listing.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut to any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

Browse Textures command (Modeling menu)

The textures browser is a specialized dialog box that provides several texture-handling functions in one location. The Browse Textures command is available when at least one texture is defined in the current model, either by creating a new texture directly or by loading a texture from the Textures folder or another location.



When the textures browser appears, a list of the textures in the current model is displayed for selection. One or more textures may be selected from the list at one time. Once a selection is made, the available operations that may be performed are indicated by which buttons in the dialog are active.

Some of the functions provided in the browser dialog are duplicates of other Textures palette buttons. They are also provided here for convenience. Some operations are available only through the browser dialog box, such as saving a texture as a separate file or deleting a texture from the active model.

- **Edit:** Clicking this button opens a texture dialog displaying a selected texture for editing purposes. Choosing this operation also closes the browser dialog box.
- **New:** Click this button to open a surface texture dialog for creating new textures.
- **Rename:** Clicking this button allows the name of a selected texture to be changed. You may also rename a texture while in its texture editing dialog, but this feature can be more convenient at times.
- **Delete:** Clicking this button deletes the selected texture(s). Since this action is not reversible through the Undo command, an alert is displayed warning of the consequences before any texture is actually deleted. A Cancel button is provided as a way to exit gracefully without taking any action.
- **Load:** Clicking this button loads a texture into the active model. A directory dialog opens pointing at the Textures folder by default. You can select one of the textures found there or use the controls in the dialog to locate a texture file in another folder.

Note that the Load Texture dialog only recognizes files with the "VisA" file type.

- **Save:** Clicking this button saves the selected texture(s) as individual files used for textures libraries. This file has the "VisA" file type designation and is recognized by the Load Texture dialog.

  During the save operation, Vision 3d renders and includes a preview image of the texture which is used in the Load Texture dialog box and on the Textures palette. To render an anti-aliased preview image, hold down the Option key while clicking on the Save button.
- **Done:** This button closes the browser dialog box and returns to the model.

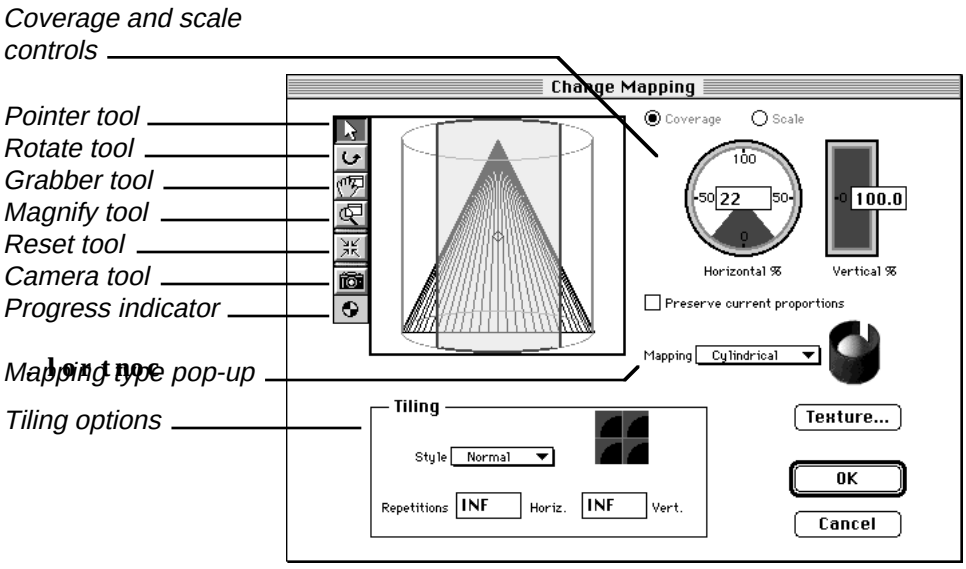
Change Mapping dialog

Applying the texture is only the beginning. The way a surface texture is wrapped onto an object is determined first by the direction from which it was applied and second by the settings in the Change Mapping dialog.

The Change Mapping dialog is opened via the Change Mapping button on the Button bar. Clicking on the button opens the dialog.

If the Change Mapping button is not active, it may be for one or more of the following reasons:

- **No object is selected.**
- **More than one object is selected.**
- **The object selected does not have a texture applied.**
- **The selection is an instance of a shape and the texture was applied on a different level, thus it is not controlled at the current level.**
- **The objected selected has a texture applied (at this level), but does not require mapping control.**



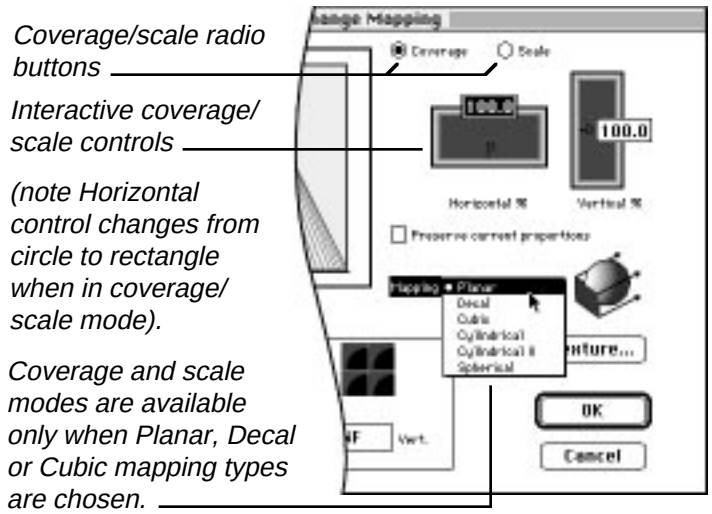
As mentioned earlier, solid textures also have mapping considerations, but do not require all of the controls in the Change Mapping dialog. This is because mapping type and tiling style are not an issue with volumetric / procedural maps. However, scale and position are important, so the Change Mapping button is available for limited functions on solid textures.

COVERAGE VS. SCALE

The Change Mapping dialog is a full-featured control panel for setting all the variables required to “fit” a surface texture to an object. The most important part of this dialog is the Mapping Preview area. This preview allows you to interactively position, rotate, and scale the applied texture to fit the selected object.

The selected object appears in wireframe form in the background with the same orientation that was used to apply the texture in the active view. A red marquee appears representing the surface map(s) contained in the applied texture.

The scale of the texture marquee in the mapping preview indicates its actual size relative to the object. This is important visual information as you decide how to wrap a texture onto an object, and at what scale.



Note the two radio buttons, titled “Coverage” and “Scale,” at the top right of the dialog box. Depending on which you choose, you may set the size of the texture as a percentage of object coverage or as a percentage of actual size. In many cases, the functional difference between the two modes is minor. But if you have taken the time to work out the exact size you want for the maps, the scale mode may be preferable.

Using the tools in the Change Mapping dialog

Tools are provided to assist you in the interactive fitting of the texture in the mapping preview area. They operate as follows:

- **Pointer tool.** Use this tool to move the texture relative to the object. Double-clicking on this tool re-centers the texture on the object if it has been shifted.
- **Rotate tool.** Use this tool to rotate the texture on the object. You may rotate it on any axis to change the texture’s orientation from the one defined by the active view when the texture was originally applied to the object. This is the same functionality as using the Rotate tool with the Control key down in a model window. The object does not rotate, but rather the texture rotates on the object.

Double-clicking on this tool resets the texture rotation back to the object's default orientation. Note that this may not be the same orientation you originally set in the active view.

- **Grabber tool.** Use this tool to scroll the mapping preview area vertically, horizontally, or diagonally to see another portion of the object. Normally, the entire object is visible within the mapping preview area. However, when the Magnify tool is used to zoom in on an object in this preview area, you may need to move around to see how the texture is fitting.

  Double-clicking on this tool re-centers the object in the preview area.

- **Magnify tool.** Use this tool to zoom in or out in the mapping preview area. This is useful for checking exact placement of a texture on an object. Each click magnifies the preview area 2X. With the Option key down, each click reduces the view scale 2X.

  Double-clicking on this tool fits the entire object and texture in the preview area again.

- **Reset tool.** Use this tool to re-set all texture transformations back to the default settings. The texture is re-centered, un-rotated, un-scaled, etc. This is useful when you wish to start over while setting the mapping parameters.

- **Camera tool.** Additionally, you can render a preview image of the texture in this preview area by clicking on the Camera tool. To initiate a preview rendering, click once on the camera tool. The "beach ball" activity indicator spins as the texture is rendered in this view.

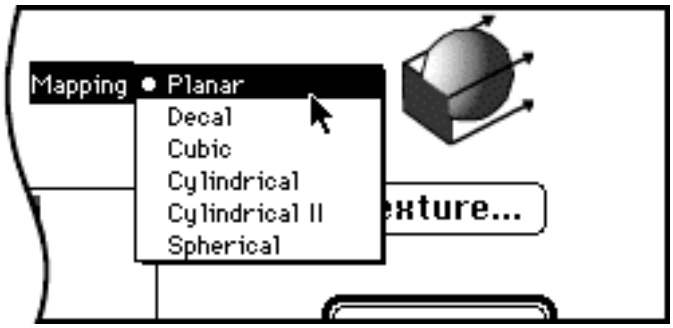
  You do not have to wait for the preview rendering to complete before making changes. If you make a change, just click on the Camera tool again to start the preview rendering over again, but using the new settings.

  To cancel a preview rendering, just select another tool. The preview will return to the wireframe display.

MAPPING TYPE

To chose the mapping type, click on the pop-up menu of possible choices. Each type of mapping applies the chosen texture containing surface maps in a different way.

Choose the best mapping type for the object.



The coverage or scale fields presented in the Change Mapping dialog vary with the mapping type selected. Surface textures may be mapped in one of the following ways:

- **Planar.** Pushes the texture straight through the object from the direction it was applied. All faces of the object show the texture, even the inside and back.
- **Decal.** Applies the texture to only one side of an object. The pattern is similar to Planar, but is visible only on the surface(s) facing the active view at the time the texture is applied.
- **Cubic.** Places a complete representation of the texture on the six faces of a cube. If the object is not a cube, so that there is no corresponding geometric faces to receive the texture, the texture is still applied from six directions and shows on the object as the map intersects the surface.
- **Cylindrical.** Wraps the texture around an object in a cylindrical fashion. If the texture is longer than the object (top and bottom) and endcaps are present, the texture smears to the center of the object on the top and bottom.
- **Cylindrical II.** Wraps the texture around the object in the same cylindrical pattern as Cylinder does, except that the end caps are not smeared. Rather they receive their own face similar to the way cubic mapping would work.
- **Spherical.** Wraps the texture around an object in a spherical fashion, with the texture gathering together at the top and bottom "poles."

When a texture is applied to an object, Vision 3d tries to determine the most appropriate mapping type for the given object type. Thus, applying a texture to a sphere results in spherical mapping as the default; a cube is given cubic mapping, etc.

However, the mapping style chosen by the application can be overridden as desired through this dialog. Any mapping type may be applied to any object. For best results the mapping style should usually be matched as closely as possible to the object's actual geometry.

Planar Mapping

Decal Mapping

Cubic Mapping

Cylindrical Mapping

Cylindrical II Mapping

Spherical Mapping

Solid (Non-map and procedural textures only)

As stated several times throughout this chapter, textures are applied to the object from the direction of the active view. This is important when orienting surface maps. Each map in a texture is centered on the texture origin point in the active view, then wrapped around the object from that point in the pattern specified by the mapping type. The texture origin point is visible as a dot in the center of the texture as displayed in the Change Mapping dialog's preview area.

TILE SETTINGS

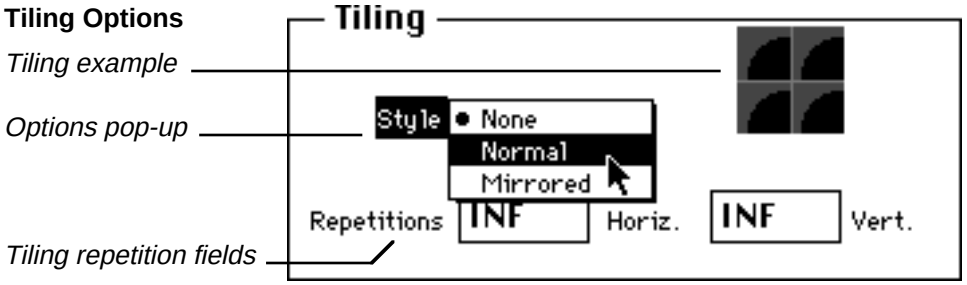
Scale and tiling are surface texture issues. Surface maps are finite in size and must either be stretched, shrunk, or tiled to completely cover an object's surface. You also want to anticipate whether the tiling pattern should be normal or mirrored across the surface. If you choose normal tiling, it is important that the maps match appropriately at the seam, otherwise the tiling will be obvious.

Mirrored tiling can minimize a tiling problem when the pattern doesn't match.

Included in the Change Mapping dialog are fields used to size an applied texture's surface maps, to set tiling characteristics, and to designate the number of surface map repetitions if tiling is enabled.

In combination with the scale and coverage settings, two different tiling patterns are provided. You may also select no tiling. If no tiling is chosen, the surface is mapped once only at the indicated scale. The map is centered on the object's texture origin point, facing the view from which it was applied.

Tiling Options



Normal tiling repeats a surface map exactly as it is oriented for the number of times specified. Mirrored tiling alternates a map end-for-end across the surface, thus minimizing the visibility of a tiling seam.

Map Repetitions

The number of times a map will be repeated is also adjustable in this dialog. The default setting is "INF" or infinite in both the horizontal and vertical directions. The infinite setting insures that an object will be completely covered no matter how large or small the map is scaled relative to the object.

A value less than infinite in these fields repeats the map a specific number of times horizontally or vertically across the surface or until the edge of the object is reached, whichever occurs first.

Mapping variations

Here are some examples of how these Change Mapping settings affect the way a surface map appears on an object. The surface map used was a simple checkerboard pattern, as shown in the following illustration.

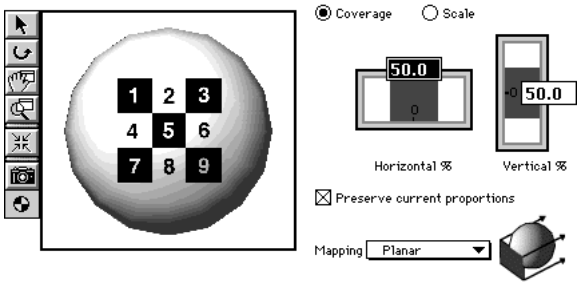
This pattern was used as an image map for the mapping examples presented below

The texture containing this surface map was then applied once to the object in the examples from the active Front view. The variations shown came from changes in the Change Mapping dialog fields only.

These mapping examples are just some of the possibilities. You may notice that no example of mirrored tiling is presented. In this case, the mirrored tile setting looks identical to the normally tiled pattern, except for the numbers.

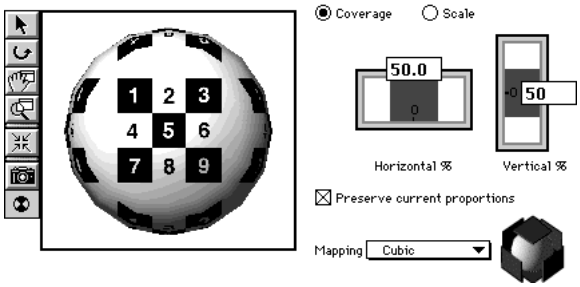
Example #1

Planar mapping, no tiling, 50% coverage



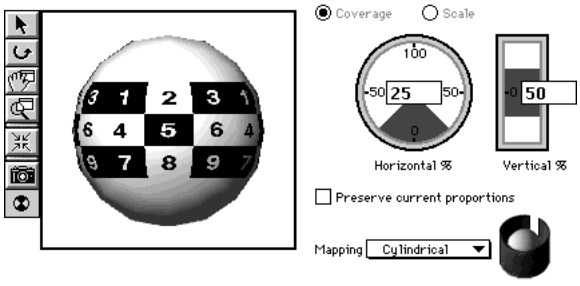
Example #2

Cubic mapping, normal tiling, 50% coverage, 1 repetition horizontally and vertically



Example #3

Cylindrical mapping, normal tiling, 50% vertical and 25% horizontal coverage, Infinite repetitions horizontally and 1 vertically



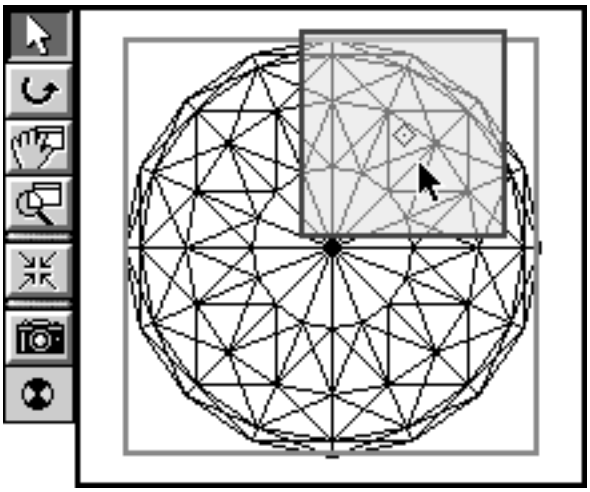
You may want to experiment further on this theme to gain a better understanding of how the Change Mapping dialog settings control the way textures can be mapped on an object. You should create a distinctive pattern such as the one used here, possibly including numbers in each of the squares. Adding numbers can tell you a lot about the positioning and pattern repetition of surface maps.

TEXTURE TRANSFORMATION

In addition to setting mapping type, scale and tiling parameters, the Change Mapping dialog allows you to control the position of the texture and its orientation.

Moving the texture

A texture may be moved or re-positioned on an object. Moving the texture marquee relative to the object in the mapping preview shifts the placement of the texture on the object in the model space so that the center of the surface map(s) is not at the geometric center of the object.



Rotating maps on objects

From time to time, it may be desirable to re-orient surface maps on an object. For example, you may want a wood grain texture to run a different direction, or prefer a specific portion of a map to be more visible.

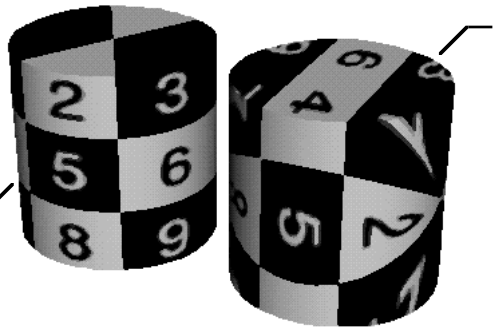
Surface maps may be easily rotated with respect to the object in two ways:

1. **Choose the rotate tool in the Change Mapping dialog and rotate the texture in the mapping preview.** Again, this is usually the preferred method because it allows you to see the results interactively.
2. **Using the object rotate tool in the model window, select the object and hold down the Control key while performing a rotation operation.** This rotates the surface map around the object, but leaves the object in the same position. The selection handles work the same for rotating surface maps as for rotating the object.

   When holding the Control key down, all other rotate tool modifier keys also apply to rotating surface maps, just as they normally would for rotating an object. You should be aware that this also changes the direction a surface is mapped to the object, as shown in the following illustration.

Rotating an image map also rotates the mapping type orientation as well

Original Cylindrical map



Same cylindrical map with texture rotated 90° clock-wise in Front view

Surface maps may be rotated on any axis. The texture does not need to be re-applied for the rotation to take effect. In fact, re-applying the texture causes the surface map(s) and mapping type to revert to the orientation of the active view again.

If you desire to rotate a surface map, but not change the mapping type orientation, you must extract the surface map from the texture definition, via the Cut / Copy / Paste editing commands, then use a paint or drawing program (one that supports the desired color depth of the image) to adjust the map's orientation. You may then re-insert it into the texture.

Texture dialog

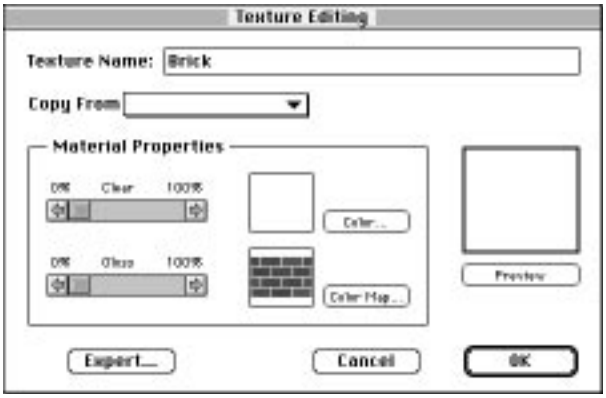
The New button on the Textures palette initiates the creation of a new texture in the current model. New textures may be created whenever a model is open.

Clicking on this button normally opens the dialogs for creating a surface-type texture. Holding down the mouse button allows you to create new solid-type textures by picking a texture extension on which the solid texture will be based.

Surface texture — Primary level

The surface texture dialog allows you to set various surface properties to describe a unique texture. A field in the dialog also provides for naming the texture, which is used in other dialogs and the Textures palette to identify the surface.

If one or more objects are selected at the time this command is called, the texture defined in the dialog box is automatically applied upon completion. Otherwise, the texture is available for manual application to objects with the Apply button on the Textures palette.

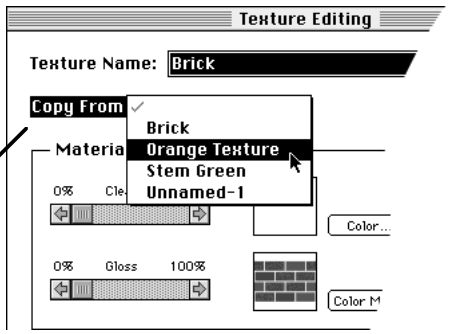


The fields presented in this dialog, with their resulting effects, are:

- **Texture Name:** This is the name the texture will be cataloged under in the textures list. It is usually best to use a descriptive name that indicates the type of texture represented.

- **Copy From:** This field allows copying settings from another texture to be used in creating a variation for a new texture. Use of this field is optional when creating a texture.

Pull down menu shows previously defined or currently loaded textures to use as a start for a new texture.



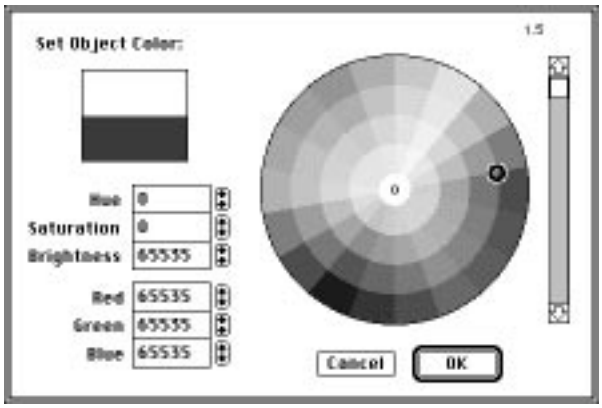
To use, position the cursor within the text box and hold down the mouse button. A pull-down menu of all textures currently in the active model will appear. Drag the cursor downward to highlight the desired texture you wish to copy. Release the mouse button to execute the copy into the new texture.

- **Material Properties:** Two slide bars are used to set the general properties of the texture. These provide a convenient, graphical way to implement the most commonly used texture properties. “Clear” sets an object’s transparency. “Gloss” sets the polished appearance. Gloss is not the same as reflectivity. The Gloss slide bar inter-connects the smoothness, reflectivity, diffuse and specular properties of a surface.
- **Color:** This field allows you to set the native color of objects using this texture. An object’s native color may be altered by other fields accessed through the “Expert” button, but it will always be a factor in creating the rendered image.

  To set the color, click the cursor in the color box to display the System color palette. The resolution of the color palette displayed is dependent on the colors available for the monitor where it is displayed.

  Position the cursor over the color desired and click the mouse button. Holding down the mouse button while moving the cursor will continuously update the values until the button is released. A color may also be set by editing the numerical values

of its components. As numbers are entered, the resulting color is displayed in the box above the fields.



The color chosen will be automatically transferred to the Texture dialog by clicking “OK” or pressing the Return key.

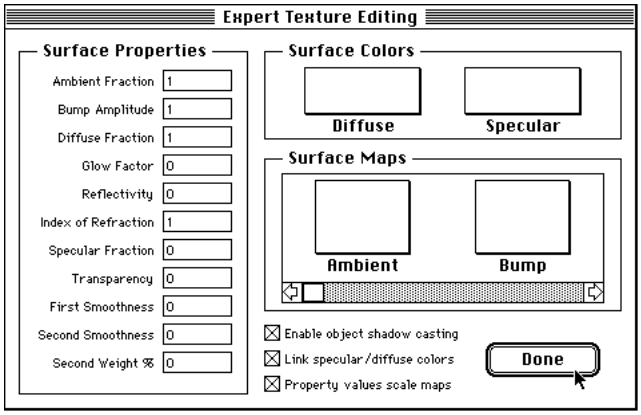
Selecting “Cancel” will exit the color palette without altering the previous color.

- **Preview:** Clicking on this button will generate a representation of the texture on a generic object.
- **Expert:** Click on this button to display a secondary dialog with more fields relating to texture creation. This display should only be used to add properties to the texture not available in the primary dialog.

EXPERT LEVEL

The expert-level dialog offers greater capability for editing texture settings. The parameters entered in these fields influence the texture in more exacting ways than the primary dialog’s sliding bars settings.

A limited explanation follows on what effect these fields have on the surface definition. They are discussed in more detail earlier in the chapter. Once you are comfortable with this software, you should seek other reference materials to fully understand the concept of surface mapping and the effect of these fields on the rendering capability that has been built into Vision 3d. Some reference materials are listed in the Appendix section of this manual



Surface Properties

Renderings of computer-generated three-dimensional graphics deal not only with the geometry of objects and their position and size, but also with the properties of their surfaces. The description and treatment of the surface of an object is as important as the geometric description of the object. Often, surface maps are a more elegant (and efficient) approach to the challenge of putting realistic details into a model than trying to create explicit geometry.

- **Ambient Fraction:** How much ambient light a surface reflects controls the apparent brightness of an object in situations where it is not directly illuminated with a light source, i.e. how visible is the surface when in the shadows? A zero value produces a surface that goes dark where shadows fall across it. A value of one results in a surface as bright as the ambient light level set in the Lighting palette.
- **Bump Amplitude:** This field sets the depth of depressions proscribed by the bump map during the rendering process. The higher the value, the deeper the depressions appear. However, the value must grow exponentially to achieve a constant increase. Put another way, a value of two is twice as deep as a value of one, but a value of four is required to double the depression again.

  Unlike the other fields in this dialog, the bump amplitude field must use a bump map to have any effect on the surface. If no bump map is present, no bump amplitude can be applied.

- **Diffuse Fraction:** All surfaces reflect light. Otherwise you would not be able to see the objects. Reflected light has two components — specular and diffuse. This field determines how much of the total light reflected from a surface is diffuse in nature. Diffuse light is light that scatters in all directions. Surfaces with predominantly diffuse reflections appear dull or matte in nature.
- **Glow Factor:** Glow determines the amount of luminescence a surface emits, without reflecting light from an outside source. The luminescence of a surface with a glow factor is not considered a light source and does not illuminate other nearby surfaces. The color of a glowing surface is determined by the color set in the texture, not the color of the light falling on the object's surface.

  Glow is used to give a neon-like appearance to objects, or in special situations with heavy shadows to insure that certain surfaces will not get too dark.

  Glow is expressed as a value from zero to one. Zero produces no emissivity, there is no maximum setting, but a value of one (1.0) will usually "blow out" or overpower any other surface characteristics.
- **Reflectivity:** Reflectivity is available in Phong, raytracing and RayPainting. Phong's reflectivity is limited to interaction with the current reflected environmental map. Both raytracing and RayPainting support inter-object reflectivity, as well as reflective environments.

  Reflection is the bouncing of light off a surface into the area around it and toward the viewpoint. When a computer model is illuminated, the color of the light, the reflective properties of an object's surface — as well as the native color of the object — determines the color for that surface.

  Reflectivity is a special case of specular light. This field determines the level of mirror-like reflection a surface may have. Reflectivity calculations must take into account the directionality of the surface — called the surface normal — at any given point. The amount of reflectivity observed varies directly with the angles between the light source, the surface normal, and the viewpoint.

  The reflectivity setting is established as a value between zero and one. A value of zero produces a non-reflective surface. A value of one is fully reflective.

- **Index of Refraction**: Refraction is available in Raytracing and RayPainting images. Light not only interacts with surfaces, but with volume as well. One volumetric light property is refraction, which causes light to bend or change direction whenever it travels through one medium to another, i.e. from air through glass through water and back to air.

  Refraction can be modeled in a Vision 3d texture as a single constant — the refractivity of the material — and is used to simulate curved or optically transparent surfaces such as flat glass, lenses, crystals, gems, etc.

  A value of one produces the refraction index of air. A table of some of the more common materials with their index of refraction is in the Appendix section.

- **Specular Fraction**: This field determines the amount of specular light reflected from a surface. Surfaces that reflect mostly specular light appear shiny or glossy. Where diffuse reflections scatter in all directions, giving a dull or matte appearance, specular reflection is light that is reflected in the direction of the angle of reflection. Specular reflections create highlights or “hot spots” on an object.

  The first smoothness field also plays an important part in the appearance of specular reflections. Smooth surfaces reflect light with less distortion, which appears on the object as pinpoint highlights. Rough surfaces with high specularity scatter the reflection such that the hot spot is more dispersed or spread out.

  The appearance of the specular light and the ratio between diffuse and specular reflections in the light coming from a surface provides important visual clues about the materials composition of an object.

- **Transparency**: This field describes the amount of light that passes through a surface in the form of reflected light from other surfaces. A transparent surface transmits light without appreciable scattering, so that surfaces behind it are visible.

  Transparent objects are treated differently depending on the rendering technique used. In wireframe, no surfaces are indicated, thus everything is transparent, so to speak. Hidden line and Quick shaded algorithms do not support transparency at all. Flat shading, Gouraud, and Phong algorithms use a “screen door” type of transparency in increments of zero, 25%, 50%, 75% and 100%, and use the increment that is closest to the

value set in this field. Raytracing can render continuously variable amounts of transparency matching the field's setting.

Transparency is expressed as a real value between zero and one. A value of zero describes an opaque surface. A texture with a value of one in the transparency field is fully transparent and therefore invisible except for its effect on the objects behind it, such as refraction.

Note that there is a special case relating to the transparency field. If the transparency field is set to 1.0, ALL other fields in the surface properties of the texture are completely ignored. This includes color, ambient fraction, diffuse fraction, specular fraction, etc. If you wish the other fields to also have some control over a texture's appearance — to have a transparent object also be reflective, for example — you must lower the transparency value (.99 or less). The texture will still be transparent, but you can also use the other fields to provide color, reflectivity, specular highlights, and/or surface haze.

- **First and Second Smoothness:** These two fields are used together to more accurately simulate the complex properties of metallic surfaces. A metallic surface is often both rough and smooth at the same time. Rough because of the microscopic faceting in its crystalline structure. Smooth due to the machining process used to create the shape.

  In addition, metals reflect light differently than other types of material. Metallic surfaces also have a high specular coefficient. Thus, even though a metallic surface may not have mirror-like reflectivity, it can still have strong specular highlights. These fields provide information to the rendering algorithms about how rough or smooth a surface is.

  Both fields have a range from zero to 32,767. A value near zero produces a less smooth appearance. The maximum value yields a completely uniform, machined surface. Values between zero and 1,000 are most often used in describing common metals. You should not think of these fields as a way to make surfaces rough…only less smooth.

  Non-metals should use only the first smoothness field or have a second weight percentage of zero.

- **Second Weight %:** This field determines how much the Second Smoothness field contributes to the surface definition. A value of

zero ignores the second smoothness field completely. A value of 100% uses this field exclusively and ignores the First Smoothness value.

Surface Mapping

The fields contained in this area of the Expert dialog provide access to all the different mappings that may be used on an object.

Diffuse color and specular color are the only maps that can be color. All the other maps must be grayscale. Grayscale maps are sampled at a certain frequency set in the rendering options dialog box. The value of each pixel in a map determines the level of effect for that property at that pixel in the rendered image.

For example, if a reflectivity map is applied to an object that has a 50% reflectivity setting, the most reflective areas of the map (those that are white) will not exceed 50%. To obtain full reflectivity in the surface map, the Reflectivity field must be set to its maximum value of one.

A texture may employ one or more surface maps in its definition. These maps correlate one-for-one with the Surface properties fields described here. The value in each surface properties field sets the ceiling for the amount of impact each surface map can have on a surface definition. In effect, they scale the range of the surface maps. If no map is present for a particular surface property, the value in the field for that property is applied evenly across the surface. If a map is present for a property, the values in the map are applied pixel-by-pixel in the rendered image, but do not exceed the ceiling set in the field for that property.

To access a map area, click within the desired box to display a dialog into which you may import or paste an image.

Surface Map Dialog

Surface mapping is a powerful feature of Vision 3d. It is the process of using two-dimensional, bit-mapped images and treating them as if they were the surface of a two- or three-dimensional object. A surface is mapped differently depending on the type of map applied to it. The user need only create an image — either by digital photo capture or using a paint program — then map it onto an object. Images can be modified in a paint-type application to achieve the patterns and contrast desired for the specific mapping effect.

Surface maps are used in three ways:

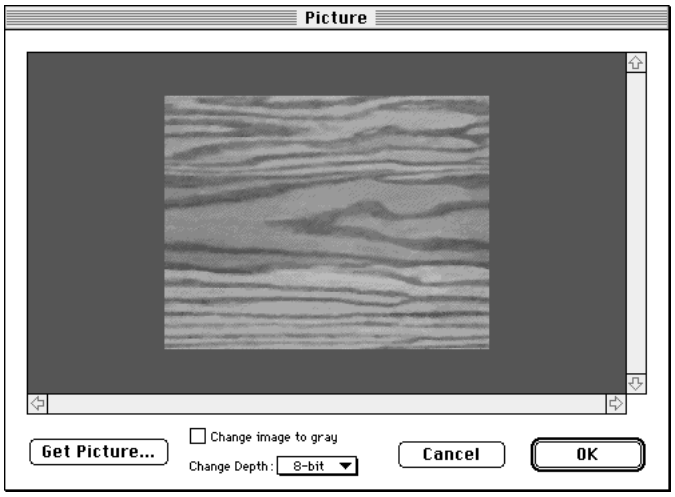
- **Bump**: This map alters an object's surface normals based on the map's grayscale values to reflect light in different directions, simulating the way light would be reflected if the object had the actual geometry described by the bump map.
- **Diffuse Color and Specular Color**: applies the color values of the map to form a detailed colored image or pattern on the surface.
- **Glow, Reflectivity, Specular and Transparency**: Uses the lightness and darkness of a grayscale image to alter the surface. The map is sampled pixel-by-pixel and the level of property represented by the map is adjusted up or down at the corresponding location on the object. White represents maximum value the property (up to the ceiling set in the properties field), black disables it. Values of gray fall within the two extremes.

Surface mapping works with the properties fields to create subtle deviations that more accurately simulate the detailed way light interacts with a surface. Mapping provides a way to include photograph-like detail — like labels, fabrics, or wood grains — to alter surfaces and give objects a more realistic appearance than is possible using geometric modeling.

For example, to create a realistic image of a brick wall, you would use a color image of the bricks for the diffuse color map and a grayscale image for the depressions of the bricks in the bump map. The result would be a rendered image that gives a very realistic "look and feel" of brick.

NOTE: *In its simplest form, mapping places a flat image on a flat plane. Other features of surface mapping allow wrapping images onto curved surfaces, such as a label wrapped around a can. In the case of a can or cylindrical shape, the image is not distorted. But a surface mapped onto a sphere or multi-sided object usually must be distorted or stretched to fit.*

Whenever one of the mapping fields is selected, the Picture dialog appears to select and load the map. A mapping field is selected by clicking on its rectangle within the Surface Mapping area.



When this dialog is open, you may load a PICT file into the dialog to be used as a map. This is done using the “Get Picture” button. You may also use the Cut, Copy and Paste commands from the Editing menu on the image in this dialog. The clipboard and Scrapbook desk accessory available through the System are compatible with the operation of this surface map dialog.

Adjustable Color Depth of Surface Maps.

Control for setting color depth is provided in the “Get Picture” dialog, which is used for all mapping fields. Images can also be converted directly to grayscale in this dialog, if needed.

Reducing a map’s color depth can save memory. For example: a 300 pixel by 300 pixel map in 32-bit color requires 264K to store as part of a texture within a model, but the same size image in 8-bit color requires only 88K. Changes to the image occur only when the “OK” button in this dialog is clicked. Once they are made, you must re-load the map into the texture again to revert to its original color depth.

While some maps need to be in full color, many others work just as well at reduced color depth. Reduced-color maps can be used successfully when the colors are simple and contain large areas of solid color. Even though the map may contain simpler colors, the

results can be just as rich. Renderings are still in 32-bit color and contain all the subtle shading and lighting-induced color shifts that give Vision 3d images their incredible realism.

Reduced-color image maps are not dithered. On occasion, you may see some banding or “artifacts” of colors in the map, especially below the 8-bit color depth. This is preferable, however, to getting pattern-aliasing from the dithering in the finished rendering. You will have to decide whether the banding (if present) is better than using more memory to load and use the map at a higher color depth.

Show/Hide Textures Palette command (Windows menu)

The Show / Hide Textures Palette commands occupy the same location in the Palettes sub-menu of the Windows menu; toggling as needed to display or hide the textures palette. The Show command is available when the textures palette is hidden. Hide Textures Palette is available when the textures palette is showing. Choosing the Hide Textures Palette command is the same as clicking on the close box of the textures palette or clicking on the Textures palette show / hide button on the Button bar.

Vision 3d displays only one textures palette, even if multiple model documents are open. This palette (and all other palettes) always refers to the active document and its current state. Switching to another model causes the textures palette to redraw showing the status of the second model, as it becomes the active model.

The Textures palette's show / hide status when a new session is started is determined by the previous session. If the palette was showing when the application was quit the last session, it will be showing when a model is opened in the current session. If it was hidden at quitting last time, it will be hidden on startup this time…the status is retained from session to session.

CHAPTER 6:
MANAGING LIGHTING

In Vision 3d, light sources are used to provide illumination. Illumination is as fundamental to 3-D modeling as any other aspect of the software. At its most basic, there must be light so that objects will be visible in a rendered image. Beyond that, lighting effects can be used effectively to add impact and enhance realism — for both still images and animations.

Objects are only visible when they are illuminated. As illumination from the light sources interact with object surfaces, the renderer determines the amount and type(s) of light reflecting from those objects. The nature of that reflected light provides important visual clues about the materials used to make those objects, as defined by the textures applied to them.

The three different types of light sources allow you to customize the lighting for the specific needs of your project. Tools and controls are provided in the software to handle the light sources just as you would handle regular geometric objects.

There are three types of light sources:

- **Directional.** These are also called Global lights, which is a reference to their illumination effects.
- **Spot Light.** Shines in one direction only, with control for setting the cone angle.
- **Point Light.** Shines in all directions from the point of illumination.

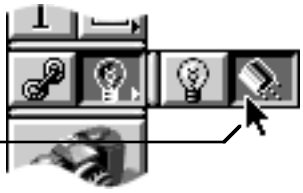
Light sources may be added to a model in any number and combination. However, as the number of light sources used in a model increases, the time necessary to render the scene also increases.

Adding multiple lights in a model also increases the overall level of illumination, with the risk of over-exposure. An automatic over-exposure control is built into Vision 3d. This feature allows for some amount of exposure latitude, but can only handle so much illumination before being overpowered. Lighting over-exposure and how to control it is explained later in this chapter.

Properties of lights

Two of the light sources, Spot and Point lights, are drawn in the model just like objects. When a spot or point light source tool is selected, the cursor changes appearance to that tool whenever it is within the boundary of the active view.

Selecting the spot light source tool on the Tools palette



Appearance of cursor when positioned in the active view

Spot and Point light sources are considered local light sources. They are represented by icons that can be manipulated interactively in the model space, using the Object Move, Object Rotate, and Object Scale tools.

Directional light sources, on the other hand, are considered global in nature. They are an infinite distance away from your model, much like sunlight striking the earth. Directional lights are added, adjusted for position, and deleted only through the Lighting palette.

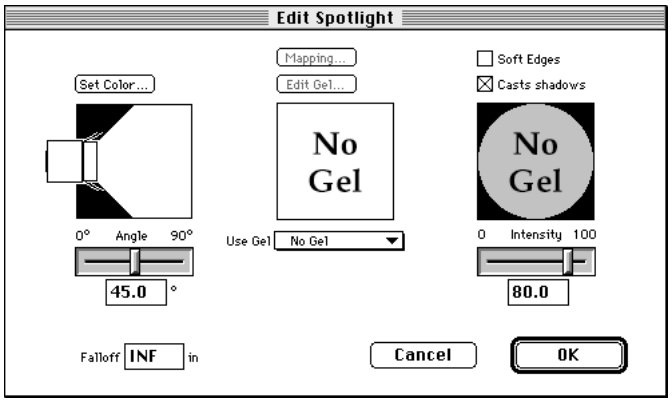
Light icons are only placeholders

The light source itself is a finite point contained within the icon. Moving and rotating the icon moves and rotates the light source inside. The size and proportion of the light's icon DOES NOT determine the intensity or spread of its illumination. That must be done through the light-editing dialogs. Depending on the type of light source, the Edit Primitive command provides different dialogs. The appropriate dialog will include controls for setting the parameters of that light source, including illumination intensity, attenuation distance, cone angle, etc. The Edit Primitive command is applicable to all three types of light source.

Spot and Point light sources do not show directly in the views. Their position is represented by three-dimensional icons in the modeling views, which indicate the light source type and position.

Directional light sources, by definition are not IN the model in the same sense as the local light sources. More accurately, they are placed AROUND the model.

Renderings that include a local light source within the image will not show any light beam or glare at the point where the light is positioned. Only the light sources' illumination on objects show in rendered images. Depending on the rendering options chosen, those effects may include smooth shaded surfaces, specular highlights on shiny surfaces, reflections on polished surfaces, shadows projected onto other objects, or any combination of these.



The three types of light sources provided in Vision 3d have one or more of the following properties in common:

- **Position** — Lights may be positioned at infinity, like a directional light source. Or they may be local sources with a specific location in 3-D space, like the spot and point lights.
- **Intensity** — The quantity of light being emitted from the source. Intensity settings are a relative index with a range of zero to 100. There is no comparison in units that can be related to light in the real world.
- **Color** — Expressed either as a hue, saturation and brightness or a red, green and blue triplet. A light source's color is not calculated in wavelength internally.

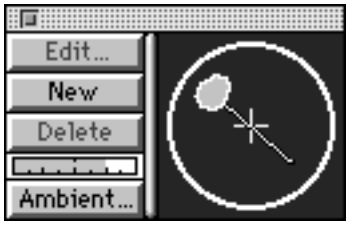
- **Gel** — Specialized surface textures can be applied to light sources to project onto the surface of one or more objects. You may use a pre-defined gel or create your own.
- **Directionality** — Lights can be omni-directional, like Point light sources or directional like Spot and Directional lights. While the position and orientation of a local light source's icon determines its direction of illumination, the size or proportion of the icon does not control or even affect the distribution of the illumination coming from that light source.
- **Spread** — The cone angle of a spot light can be adjusted. Only the spot light source has this control. You may also set a hard or soft edge on the edge of the cone where the illumination ends.
- **Falloff** — The proximity of the light to an object determines how brightly the object is illuminated when the Falloff field is set at less than "infinite." When set to infinite, there is no attenuation of a light source's intensity. The directional light does not have an attenuation control; only local light sources have this feature.

Intensity, color, cone angle and attenuation are adjusted for each light source individually through the Edit Primitive command in the Modeling menu.

Additionally, light sources in Vision 3d can accept textures — either as a Gel (within the Edit Light dialog), or as a texture applied to the light source as you would any other object (in the model window). Applying volumetric maps (in solid textures) or surface maps (in surface textures) to light sources causes them to be projected into a scene, providing realistic environmental lighting effects like multi-color washes, venetian blinds, window frames, curtains, etc.

Global/Directional lights

When a new model is started, a single directional light source is automatically present in the Lighting palette. Its default position is set so that the illumination is coming from the upper left front quadrant. (In other words, over your left shoulder as you face the screen).



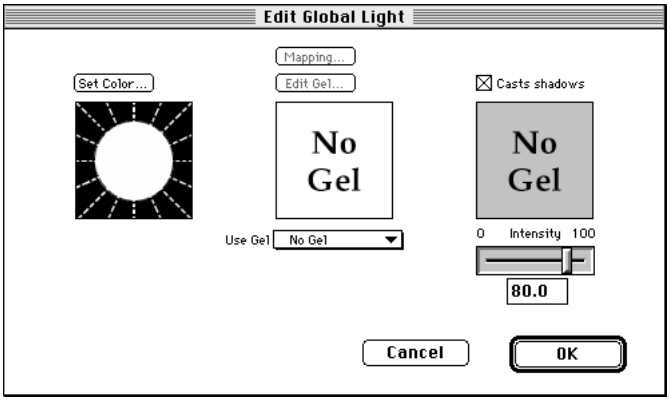
Of course, you can add, move, change intensity, or even delete the directional light sources using the buttons provided on the Lighting palette. If you remove all directional lights, the model will appear very dark when rendered unless you insert one or more local lights to take the place of the directional light.

The intensity of the default directional light is set at 80%. As you can see from the light meter on the Lighting palette, a single directional light at 80% intensity is sufficient to fully illuminate a model. The light meter measures only the cumulative amount of all directional lights in the model and the ambient light level. It does not consider local light sources.

You can add as many directional lights as you wish, although this is not a typical use of this type of light. Multiple directional light sources can over-expose a model very quickly. If you choose to do this, reducing the intensity of each light source is highly recommended.

The cumulative effect of multiple surfaces on the lighting intensity within a model must be considered in the overall appearance of a rendered image. Ideally, the total of all light reflected from objects in a model should fall in an internal range between zero and one. If the total illumination exceeds this range, the result is over-exposure. This can be avoided by giving attention to light source positioning and lighting intensities.

Directional light sources shine “toward” the model from a direction outside the model space. As mentioned, a directional light source is intended to simulate sunlight striking the earth. Its light “rays” are parallel to each other.



The illumination parameters of a directional light source are set or adjusted through the Edit Global Light dialog. This is opened by selecting the directional light on the Lighting palette, then clicking on the Edit button on the left side of that palette.

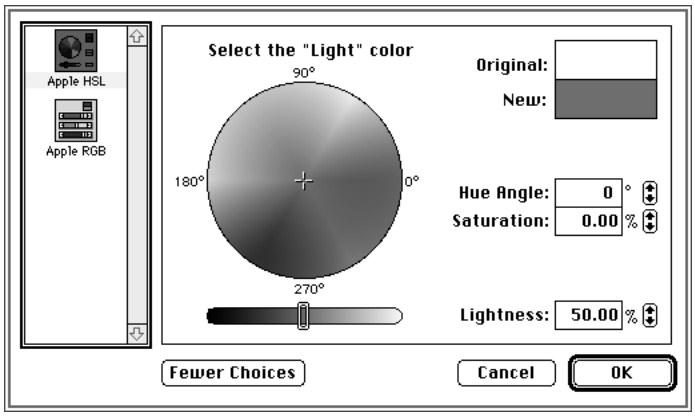
The settings available in the Edit Light dialog for directional light sources is explained as follows:

Setting color

The directional light source(s) present in the Lighting palette displays in the color given in the Edit Light dialog box. By default, directional lights are white, but may be any color you want.

The color for a directional light is set with the standard Apple Color Picker dialog box. The color picker dialog is opened by clicking on the Set Color button in the Edit Light dialog. In configurations where there are multiple monitors present on a machine, the color picker dialog ALWAYS opens on the monitor with the best color depth.

***NOTE**: Even if a monitor is turned off, but its video card is installed in a NuBus slot on the machine, the operating system still thinks the monitor is present as part of the desktop extents. Thus, if the monitor that is off has a higher color depth than the other(s), the color picker actually opens but you will not see it and the software may appear to freeze up.*



The number of colors presented for selection in the color picker dialog depend on the color depth of your monitor.

Once a color is selected, clicking on the OK button returns you to the Edit Light dialog. The chosen color is then represented in that dialog. Clicking on the OK button again in the Edit Light dialog returns you to the model.

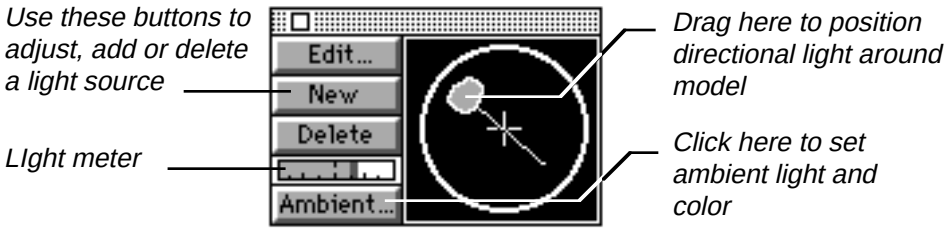
Setting intensity

A directional light's intensity can be adjusted graphically with the slider control or numerically in the field below the slider. As expected, a setting of zero produces no illumination, 100 is maximum illumination.

If you need more than 100% illumination, you can add multiple directional lights in the same location on the lighting sphere.

Lighting palette

Any discussion of directional light sources must include the Lighting palette upon which this type of light is represented and accessed. The Lighting palette presents controls for handling directional light sources, as well as ambient lighting.


Use these buttons to adjust, add or delete a light source
LIght meter
Edit...
New
Delete
Ambient...
Drag here to position directional light around model
Click here to set ambient light and color

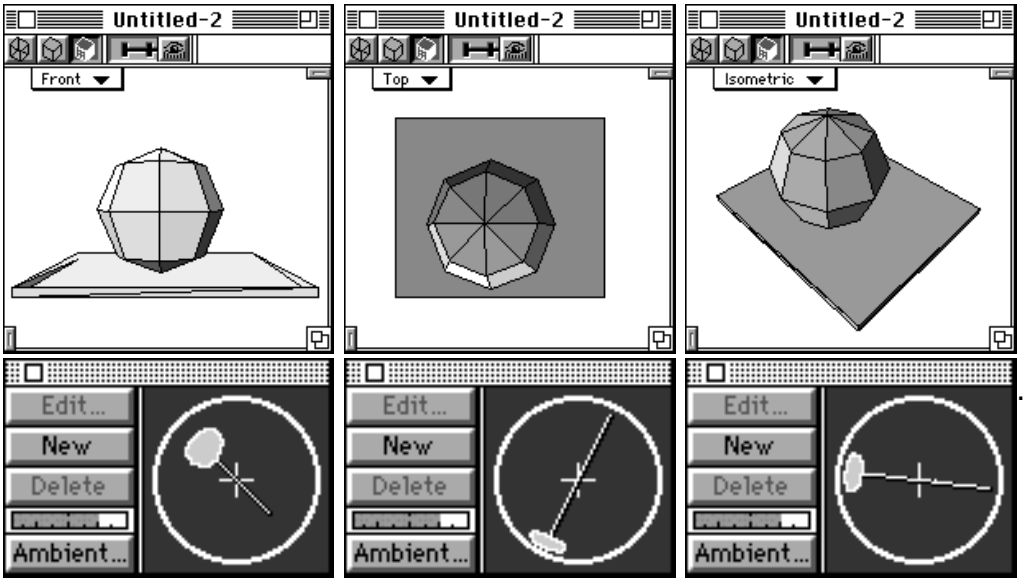
The Lighting palette is open by default the first time Vision 3d is launched. It may be hidden or shown by using the “Show / Hide Lighting Palette” command in the Palette’s sub-menu of Windows menu or by clicking on the Lighting palette’s show / hide button in the Button bar. Alternately, you can click on the palette’s close box, which has the same effect as using the hide command.

As mentioned, directional light sources illuminate a model with parallel rays of light, similar to the way sunlight illuminates objects on Earth. They are considered the primary light source for a model. In comparison, ambient lighting is a non-directional general illumination that fills shaded or shadow areas of a model so that details not directly illuminated by light sources are still visible.

Positioning a directional light on the lighting sphere

Think of the area on the palette containing directional light source icons as a sphere surrounding the model. Directional light sources may be added or deleted to the sphere as needed. Each light may be selected and moved anywhere on the sphere to illuminate the model from that direction.

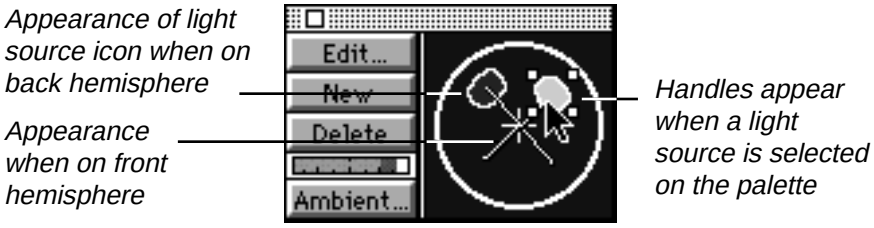
A directional light’s position may be adjusted by dragging the light source around the lighting sphere. A pointer is attached to each directional light showing the direction from which the light is striking the model with that source



The lighting sphere displays the position of directional light sources relative to the currently active view. If the active view is changed, the position of the light(s) shifts on the Lighting palette to correspond with the new view.

Toggling directional lights between hemispheres

To move a light from the front hemisphere to the back (or vice versa), based on the active view, hold the Control key down while dragging the light source. The light source changes appearance to indicate its new global position relative to the model. Releasing the mouse button before the Control key keeps it on the new hemisphere. The light reverts to its original hemisphere if the Control key is released before the mouse button.


Appearance of light source icon when on back hemisphere
Appearance when on front hemisphere
Edit...
New
Delete
Ambient...
Handles appear when a light source is selected on the palette

Once toggled to a hemisphere, you may move the light without holding the Control key down. Usually, directional lights on the back hemisphere have a reduced illumination effect. Back lighting a model with a directional light may be useful for bouncing a highlight off some of the objects. But do not expect a translucent object to glow when illuminated from behind. This effect is not supported by the rendering algorithms. Applying a texture with a glow factor is recommended instead to effectively simulate this effect.

It may take a couple of adjustments to get the highlight to reflect just right from a backlight, but does add to the quality of an image.

Light meter

The Lighting palette presents a simple light meter to warn of potential over-exposure problems. This light meter measures the cumulative light coming from the directional light sources and the ambient light level. Local light sources are not considered in the meter's display.

The meter is intended to alert you to the possibility of over-exposure only, not necessarily that it will occur. For example: The placement of a directional light at half intensity on the back hemisphere (relative to the active view) can provide a needed fill illumination, but will not over-expose.

Over-exposure is usually first noticed in surface maps that appear washed out. If you are not sure how and where excess illumination levels are affecting the scene, you can set the lighting control to "Debug" through the Lighting Effects command in the Rendering menu. In that mode, rendering an image (usually of preview quality) causes the debug color to appear wherever the illumination levels are too high. You will then be able to manually adjust the light sources as needed to compensate.

Additionally, Vision 3d provides several automatic exposure control methods to help control excess lighting. These are also accessed through the Lighting Effects dialog box. These lighting options are discussed later in this chapter under the Lighting Effects command's description.

Modifier keys that apply to Directional light sources:

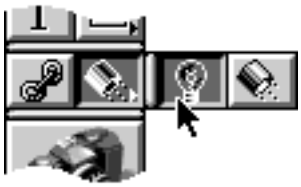
A. Shift key — Extends the selection of light sources on the palette. Multiple light sources may be selected at one time for moving in unison, setting parameters or applying textures.

B. Control key — Toggles the hemisphere orientation of a Directional light source as it is being dragged around the model. The front / back hemisphere orientation in the lighting palette is always relative to the active view.

C. Option key — Leaves a copy of the light source in the original position as the original is dragged to a new location. This is an alternative method to clicking the New button on the palette, but is most useful when you need to exactly duplicate a light source's parameters for another light.

Point light sources

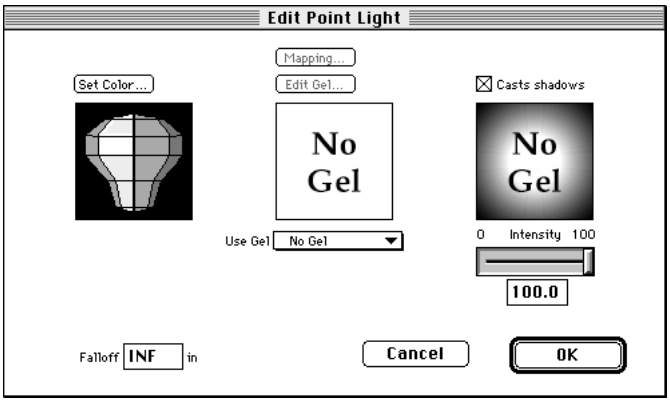
A Point light is a local source of illumination. It shines in all directions from a single point contained within the point light icon. A point light source is inserted into the active model using the Point Light tool.



When a new point light is inserted into a model, the Edit Light dialog box automatically appears so you can change its color, intensity, attenuation, gel, etc. from the default settings.

This is the same dialog that opens if you choose the Edit Primitive command in the Modeling menu when a previously-inserted point light source is selected. If you make changes to the light's settings in this dialog, clicking on the OK button accepts those changes and applies them to the light source. If you click on the

Cancel button when inserting a new point light source, no changes are made from the default settings, but the light source is still inserted into the model. To remove a point light source, simply select it and push the Delete key. Alternately, you can choose the Delete command in the Edit menu.

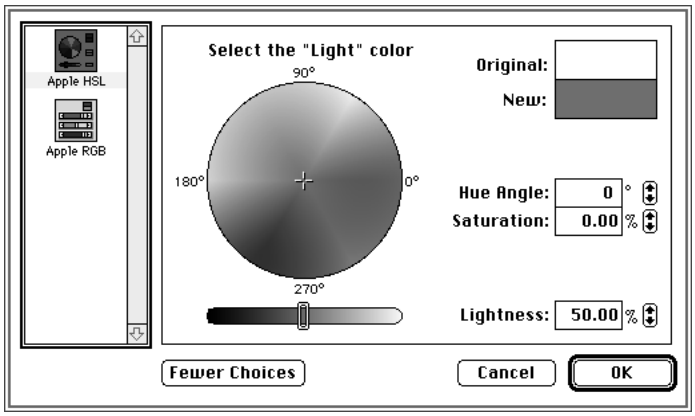


If you DO NOT want the Edit Light dialog to open for a light source upon insertion, i.e. — you want to automatically accept the default setting for the light source, hold down the Control key before clicking or click-dragging in the active view. The light source is inserted into the model with its default settings and you are not bothered by the Edit Point Light dialog.

The fields and controls found in the Edit Point Light dialog box are as follows:

Setting color

The color for a point light is set with the standard Apple Color Picker dialog box. The color picker dialog is opened by clicking on the Set Color button in the Edit Light dialog.



Once a color is selected, clicking on the OK button returns you to the Edit Point Light dialog. The chosen color is then represented in that dialog. Clicking on the OK button again in the Edit Light dialog returns you to the model.

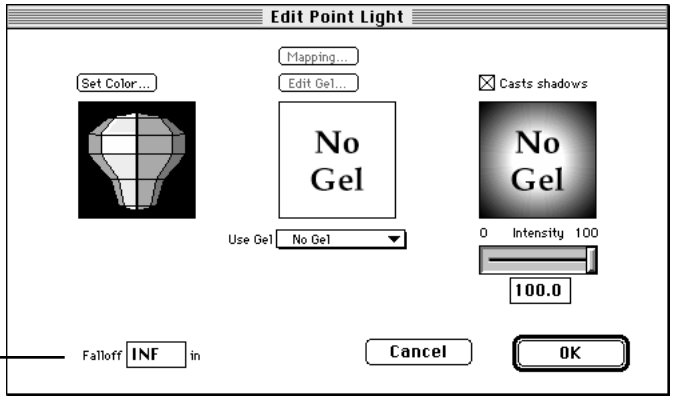
Setting intensity

A point light's intensity can be adjusted graphically with the slider control or numerically in the field below the slider. As expected, a setting of zero produces no illumination, 100 is maximum illumination for that light source.

Inserting multiple light sources is effectively the same as increasing the intensity above 100% for one light source. This is not recommended under normal circumstances because of over-exposure concerns.

Setting attenuation

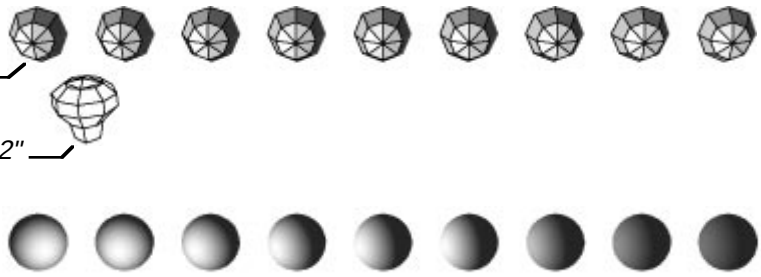
By default, point light sources are inserted in a model set to an infinite distance of illumination. For most purposes, this setting is desirable. Entering an illumination distance in the Edit Light dialog for a point light source causes its illumination to decrease over distance. This can be useful when setting up special or dramatic lighting effects in a local area of a model or scene.



This field sets the maximum distance at which a Point light source will illuminate objects

The Falloff field indicates how far the illumination travels. This value is the absolute maximum distance that an object will be illuminated by that light source. Reducing the intensity of the light source will not cause the attenuation to fall off any quicker. The falloff value is a specific distance — meaning that if the light source is reduced in intensity, it will still illuminate objects in the model to that distance, but the closer objects will be less bright.

Light attenuation



Example model

Falloff for this light source was set to 12"

Rendered image show illumination falloff, with last sphere at 13.5" not illuminated at all

There is no upper limit to the value, unless you consider “infinite” a limit. The infinite setting (INF) disables any attenuation effect, i.e. — no falloff occurs for that light…ever. This field uses the same units defined in the Set Rulers dialog box, which is the units of scale for the model being assembled. The value in this field is the distance at which a light source’s illumination is no longer visible; the distance at which the falloff of the illumination from that source is complete.

Modifier keys that apply to point light sources:

A. Shift key — Constrains the tool to draw the icon with the same dimensions on all axes.

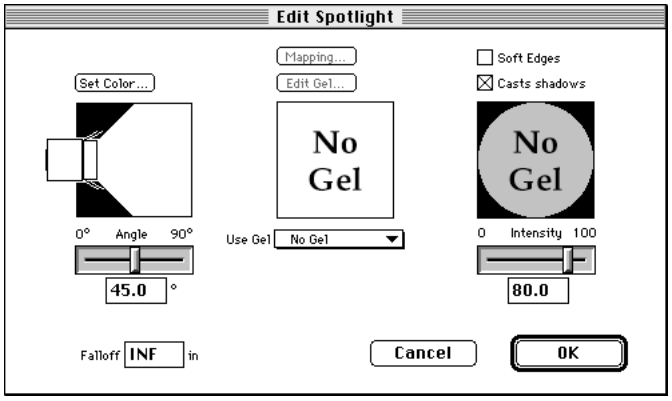
B. Option key — Draws an object from its center, instead of the corner.

Spot light sources

A spot light is also a local source of illumination, like the point light. However, it shines in only one direction, like the directional/global light. The illumination comes from a single point contained within the point light icon, so its light rays are not parallel like the directional light.

Just like the point light, a spot light source is inserted into the active model using the Spot Light tool, located just under the Text tool on the left side of the Tool palette.

When a new spot light is inserted into a model, the Edit Spot Light dialog box automatically appears so you can change its color, intensity, cone angle, attenuation, gel, etc. from the default settings.



This is the same dialog that opens if you choose the Edit Primitive command in the Modeling menu when a previously-

inserted spot light source is selected. If you make changes to the light's settings in this dialog, clicking on the OK button accepts those changes and applies them to the light source. If you click on the Cancel button while inserting a new spot light, no changes are made to the light source's default settings, but the light source is still inserted into the model. To remove a point light source from a model, simply select it and push the Delete key. Alternately, you can choose the Delete command in the Edit menu.

If you DO NOT want the Edit Light dialog to open for a local light source upon its insertion, hold down the Control key before clicking or click-dragging. The light source is inserted into the model with its default settings.

The fields and controls found in the Edit Spotlight dialog box are as described in the following topics.

Setting color

The color for a point light is set with the standard Apple Color Picker dialog box. The color picker dialog is opened by clicking on the Set Color button in the Edit Spotlight dialog.

Once a color is selected, clicking on the OK button returns you to the Edit Spotlight dialog. The chosen color is then represented in that dialog. Clicking on the OK button again in the Edit Spotlight dialog returns you to the model.

Setting intensity

A spot light's intensity can be adjusted graphically with a slider control on the right side of the dialog. You can also set the intensity numerically in the field below the slider. As expected, a setting of zero produces no illumination, 100 is maximum illumination.

If you need more than 100% illumination, you can add multiple spot lights in the same area of a model. This is not recommended, because of over-exposure concerns. However, you may choose to use multiple spot lights to achieve soft edges on the shadows. In this case, reducing the intensity of the lights is highly recommended.

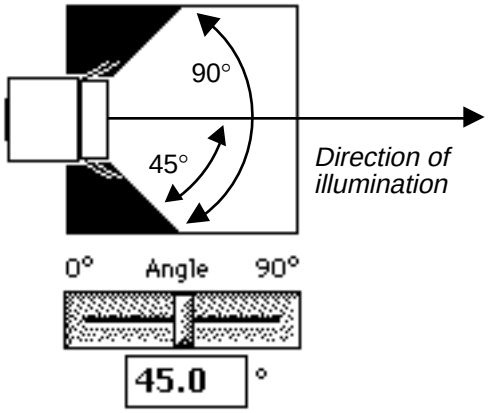
Setting cone angle

Only the spot light source has a cone angle setting. A spot light's cone angle or degree of spread can be adjusted graphically with a

slider control on the left side of the dialog. You can also enter the cone angle numerically in the field below the slider. The range is from 0° to 90°. Of course, a setting of zero shuts off any illumination coming from the spot light, regardless of the intensity setting.

Measuring cone angle

A setting of 45° for the cone angle in a Spot-light's Editing dialog corresponds to an included angle of 90°; 60° corresponds to 120°; 90° corresponds to 180°; etc.



The cone angle is measured as the amount that the cone of illumination diverges outward from the direction of illumination. This is different than the total angle of the cone of illumination. That is, a setting of 45° has an effective total angle of 90°, a setting of 90° is effectively 180°, and so on.

Obviously, the size of the cone of illumination striking objects in the model depends on both the cone angle and the spot light's distance from the object. Just as in the real world, a spot light can be moved closer to or farther away from an object to adjust the size of the spot of light.

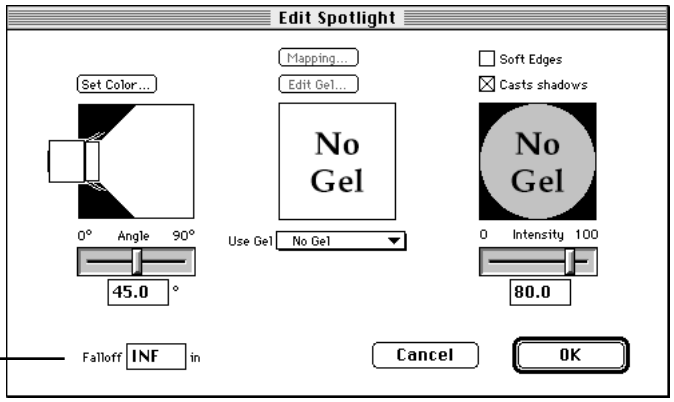
Setting soft edges on the spot light

Connected with the cone angle, there is a checkbox in the spot light's editing dialog that allows you to set hard or soft edges on the cone of light as it strikes objects. This is located in the upper right corner of the dialog.

The default is to illuminate with hard edges on the spot of light. Turning on soft edges requires more calculations and can slow the rendering speed, especially if many spot lights are used in a model.

Setting attenuation

By default, all light sources are inserted in a model set to an infinite range of illumination. For most purposes, that setting is desirable. Setting a falloff distance in the Edit Light dialog for a light source causes its illumination to decrease over distance. The ability to do this can be useful when setting up special or dramatic lighting effects in a specific model or scene.



This field sets the maximum distance at which a Spot light source will illuminate objects

The Falloff field sets the maximum distance a light source will illuminate objects. Reducing the intensity will not cause the attenuation to fall off quicker. The attenuation distance is a specific measurement, meaning that if the light source is reduced in intensity, it will still illuminate objects in the model to that distance, but the interim objects will be less bright.

There is no upper limit to the value, unless you consider "infinite" a limit. The infinite setting (INF) disables any attenuation effect — no fall off occurs for that light. This field uses the same units defined in the Set Rulers dialog box, which are the units of scale for the model being assembled. The value in this field is the distance at which the light's illumination is no longer visible; the distance at which falloff of the light from that source is complete.

Modifier keys that apply to spot light sources:

A. Shift key — Constrains the tool to draw the icon with the same dimensions on all axes.

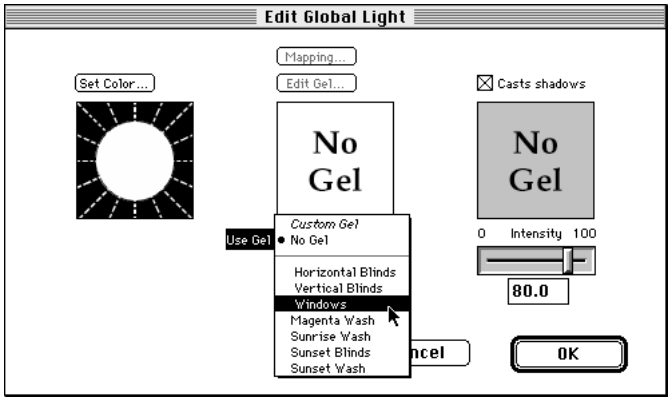
B. Option key — Draws an object from its center, instead of the corner.

Applying gels (all light sources)

Special textures may be applied to any light source. These are called “Gels.” When applied, the light source projects the pattern or image present in the gel onto the object(s) that the light source is illuminating.

Built-in gels

Several gels are included with the Vision 3d package. They are stored in the Gels folder within the StrataVision 3d application folder. Each time Vision 3d is launched, this folder is scanned (along with the Extensions, Textures, and Shapes folders) so a list of available gels can appear in a pull-down menu within the Edit Light dialog.

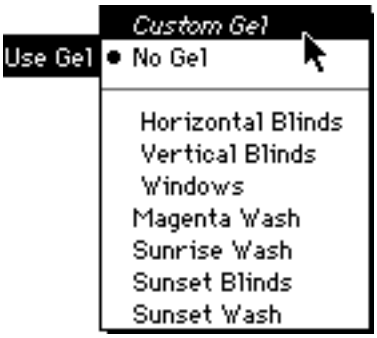


To apply a gel to a light source, click on the Use Gel pull-down menu. The gels available during this session appear in this menu. Choosing one from the list automatically loads the gel and applies it to the light source. That gel’s image becomes part of the model and is saved as part of the document. When you render the model, that light source will project its illumination through the gel you applied. If you change gels, the previously chosen gel is deleted from the model.

As mentioned, a gel is a specialized texture. It contains not only the pattern or surface map to be projected, but also the mapping and scale information that fits the texture to the light source in a manner so it will work correctly the first time. In other words, all the work has been done for you, you just have to use the gels provided through the Edit Light dialog.

Custom gels

There will be times when you want to have a light source project an image of your own selection. Choosing “Custom Gel” in the Use Gel pull-down menu allows you to create a new surface texture to be used as a gel for the selected light source.



The familiar surface texture editing dialog opens. Typically you would load a color map into this texture to be projected by the light source. You can also use the expert mapping fields to affect the illumination coming from the light source. The same principles discussed in Chapter 5 — using grayscale maps to control surface properties on objects — is also applicable to gel creation.

Thus, diffuse and specular maps can also be an effective method of changing illumination intensity in specific patterns that simulate how a light would cast shadows as it shines through a window, door, translucent panel, or other object. Or perhaps you want to project a graphic design onto an object with another texture applied, as shown in the previous illustration.

Of course, you can get the same result by creating a new texture through the New Texture command (or New button on the Textures palette) and applying it to the local light source manually in a model window. The advantage of using the Use Gel field in the Edit Light dialog is that the texture is automatically applied from the appropriate direction to be projected by the light source. Manually applying a texture to the light source icon requires that you set the application direction yourself, as you would with any other object.

The Edit Light dialog's "Custom Gel" feature allows you to use only surface textures as gels. If you want to use a solid texture on a light source, you will need to create or load the texture and apply it to the light source manually. An example of using a solid texture on a light source would be to apply a Rotoscope texture to play a PICS file or QuickTime movie so the light acts as a movie projector in an animation.

In any case, you would still have to set the mapping style and texture scale. If you do apply a texture in the conventional way, note that when you open the Edit Light dialog for that object, that texture will be represented there as a custom gel.

Quick tutorials

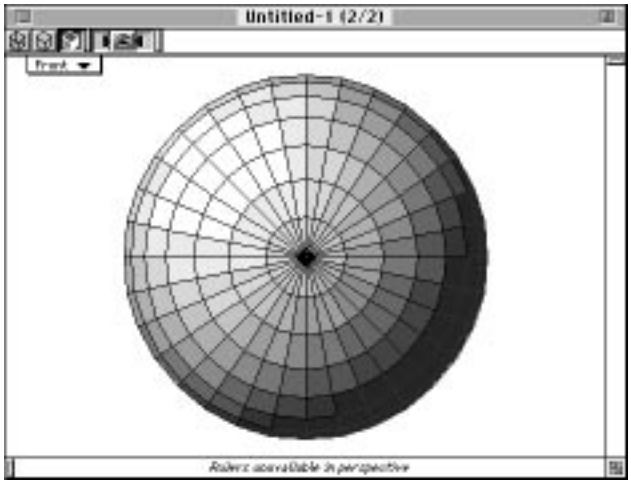
The object of these tutorial exercises is to help you understand how light sources are placed into a scene and edited, and how different types of lights and lighting techniques can be used to achieve the effects you want to see in a finished rendering.

Exercise 1: handling lights

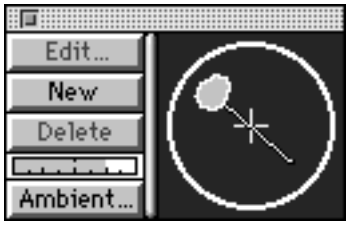
In Exercise 1, you will learn how to place light sources in a scene and how to reposition them. You'll also see how directional lights and spot lights illuminate a scene differently and how ambient light provides "fill" lighting.

1. **Create a new model.** As before, insure that the lens is set to orthographic and the view is set to Front.

2. **Draw a large sphere in the model window.** The size is not important, just make it a comfortable fit in the model window.
3. **Change the three-position lens control to Normal.** Vision 3d redraws the sphere in perspective at a slightly smaller size. Use the "Fit Views to All" command to fit the sphere to the model window again.
4. **Change the Display Method to Shaded Surfaces.** Vision 3d again redraws the sphere, but this time with shading. Notice how with the Shaded Surfaces display method you can see light from the default directional light source shining on the top left portion of the sphere.

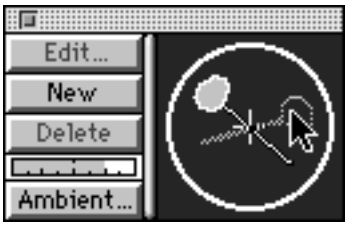


Look at the Lighting palette (usually positioned in the lower left corner of your screen). There you see the single default light source is positioned to shine down on your model from the "10 o'clock" position.



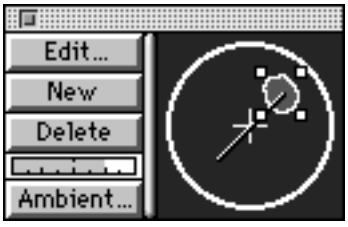
5. **Reposition the light source.** Place the cursor on the light source as shown. Click-and-drag to move the light source to a new

position within the lighting circle. When you release the mouse button, Vision 3d redraws the scene with the light source illuminating the model from its new position. Reposition the light source several times. Notice how each time you reposition the light source, the lighting in the model changes with the new position.



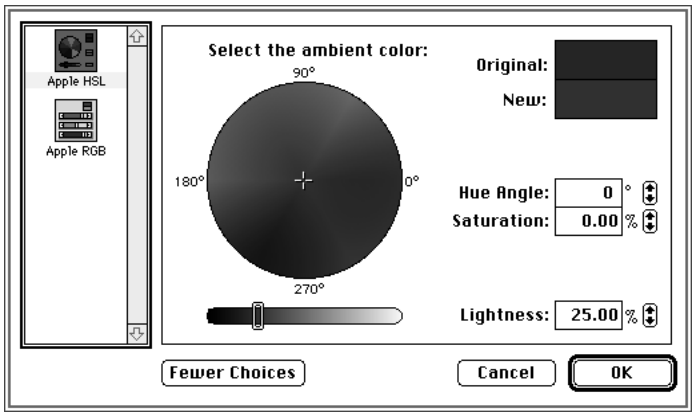
The circle on the right side of the Lighting palette represents an imaginary sphere that encompasses your model. By positioning a directional light source at a specific point on this imaginary sphere, you indicate to Vision 3d from which direction the light source should illuminate your scene in the model window.

6. **Reposition the light source in the rear hemisphere.** Hold down the Control key while you drag the light source. Notice how this moves the light source to a similar position behind the object in the model window.



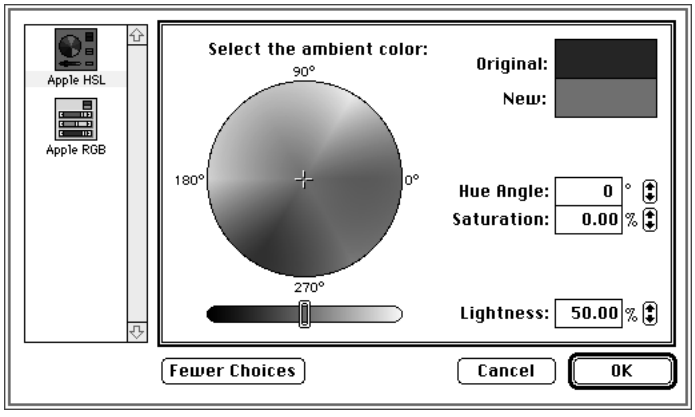
Holding down the Control key while repositioning a directional light source toggles between the front and rear hemispheres of the lighting sphere, relative to the active view. Now reposition the light source again, but this time don't hold down the Control key. Notice how the light source remains in the rear hemisphere. It will stay there until you toggle back to the front by holding down the Control key while dragging the light source.
For now, leave the light source in the rear hemisphere. Notice how the front of the sphere appears dark, since no light source is currently illuminating that part of the sphere.

7. **On the Lighting palette, click on the Ambient button in the bottom left corner.** Vision 3d opens the Apple Color Picker dialog box and displays the current color of ambient light in the scene. The actual layout of the color picker displayed differs with the color depth of your monitor, but they all work essentially the same.



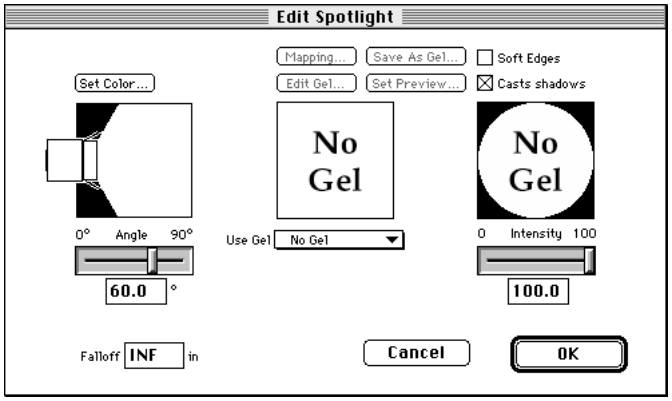
Notice that the ambient light brightness is quite low, but not black. The setting is for a neutral, dark gray color, as displayed in the upper left corner of the dialog window. *This is the recommended setting for ambient light.*

8. **Move the brightness slider in the color picker dialog until it is positioned about midway between the black and white.** Notice the new color you've specified is significantly lighter than the default setting.



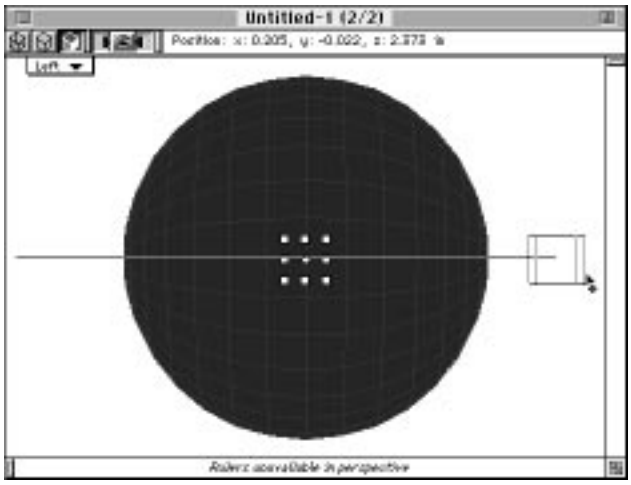
Ambient light is non-directional, general illumination that provides “fill” light for dark areas of a model. Ambient light has no source; it’s just present in the model. The level of ambient light is controlled through this color picker — the brighter the color, the brighter the ambient lighting in the model.
You may also pick a color other than different levels of gray for the ambient light. This can be useful for special lighting effects in the shadow areas of a rendered image.

9. **Click on the OK button.** Vision 3d redraws the model window with the new level of ambient light you’ve defined for your model. Notice how your model appears brighter now than it did when the ambient light was set to the darker default color. Notice also that the light meter on the Lighting Palette is now into the red, warning of possible over-exposure or too much light in your model. To avoid possible over-exposure, you must reduce the level of ambient light to what is was before.
10. **Return the ambient light level to approximately the default.** Refer to steps 7 and 8, if needed, to help you recall how to edit the ambient light level. Move the slider back to a position near the default setting and click on the OK button. Vision 3d will redraw the scene again with the re-edited ambient light level.
11. **Replace the directional light source with a spot light source.** To do this, first delete the directional light source. On the Lighting palette, click on the directional light source to select it. You know it’s selected when four selection handles appear around the head of directional light. With the directional light source selected, click on the Delete button on the lighting palette. Vision 3d redraws the model without the directional source.
Notice how dark the model now appears.
12. **Place a spot light in the scene.** To do this, choose the Spot Light tool from the Tool palette. Notice how the cursor icon changes to a spot light icon when moved into the model window. Position the cursor near the center of the sphere and click once to place the spot light into the scene.
Vision 3d opens the Edit Spotlight dialog box. This is where you can set the cone angle, color, intensity, and other parameters for the spot light. Click on the OK button to accept the spot light’s default settings. If you wish to re-open this dialog for a selected spot light, choose the Edit Primitive command from the Modeling menu.

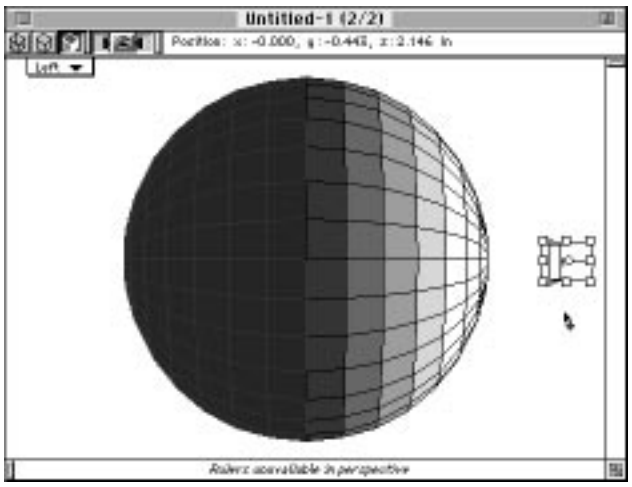


Vision 3d redraws the model with the spot light's icon visible in the modeling window.

13. **Reposition the spot light source.** With the spotlight source selected, switch to the Left view. Notice that the light source was inserted at the same depth from the Front view into the model. It is occupying the same space in the model as the sphere.



Choose the Object Move tool from the Tool palette, then click on the right center selection handle and drag the spot light to the right edge of your model window. When you click on the selection handle, a line from the spot light indicates which direction the light is shining.



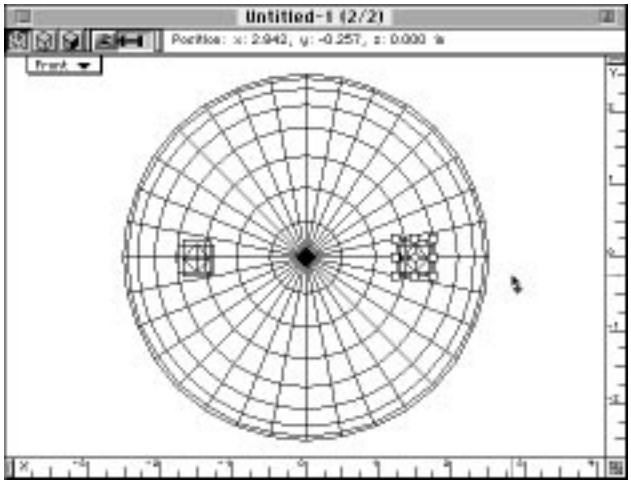
After the light source is repositioned, Vision 3d redraws the scene again. You can now see the light from the spotlight source illuminating the front of the sphere, as viewed from the left side. By changing back to the Front view, you can see how the spot light is casting an illumination circle on the sphere.

This exercise was intended as a simple introduction to handling light sources. No rendering was required, because the effects of moving light sources was obvious enough in the model window. More advanced features of the light sources in Vision 3d may not show until the model is rendered. Some of these advanced features are presented in the next exercise.

Exercise 2: Using gels

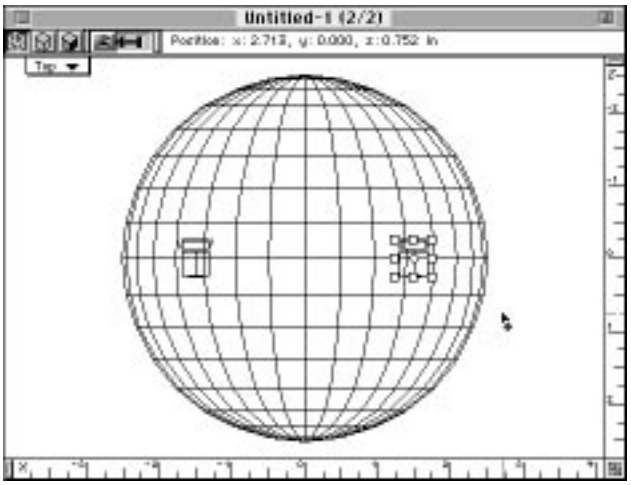
In this exercise you'll learn how to apply colors, textures and gels to light sources and how to use light sources to project a pattern onto the surface of an object. You'll also learn how to edit light intensity and other lighting parameters.

1. **Create a new model.** As before, make sure the lens is set to Orthographic and the window is displaying the Front view.
2. **Draw a large sphere.** Use the Transform command to set its absolute size to 5" in diameter. Also set the sphere's position to absolute zero on all three axes while in this dialog. Use the "Fit Views to All" command in the Windows menu, if necessary, so you can see all of the sphere in the view.
   In this exercise, size is important because multiple light sources will be illuminating different portions of its surface. It must be large enough to minimize overlap of the illumination.
3. **Place two spotlight sources into the model.** Position them on each side of the sphere, as shown in the following illustration. Refer to the previous exercise if you need a reminder on how to insert spotlights into a scene.



When you insert each spotlight, Vision 3d opens the Edit Light dialog box. Just click on either the OK or Cancel buttons both times when this dialog opens. The spot lights are inserted using the default settings.

4. **Reposition both spotlights outside the sphere.** If you switch to the Top view, you will see that the spotlights were inserted in the model at the same depth as the sphere, so they are actually inside the sphere.



NOTE: *You will have difficulty selecting either or both of the spotlights, because they are inside the sphere. There are several ways to select objects inside or behind other objects. You may want to review the sections in chapter 1 on commands provided in the Selection sub-menu in the Edit menu and in chapter 3 concerning changing selection criteria in the mouse filter dialog.*

One method of selecting objects inside or behind other objects is by changing the selection criteria. Choose the Mouse Filter command from the Selection sub-menu (in Edit menu). Vision 3d opens the Mouse Filter dialog box. Make sure the Edge checkbox is enabled, and the Surface checkbox is NOT enabled. Click on the OK button. Vision 3d returns you to the model window. Now click on the edge of a spotlight to select it. By adjusting the selection parameters, you can define alternate selection criteria for the cursor, making it easier to select objects that are positioned very close to, or inside of, other objects.

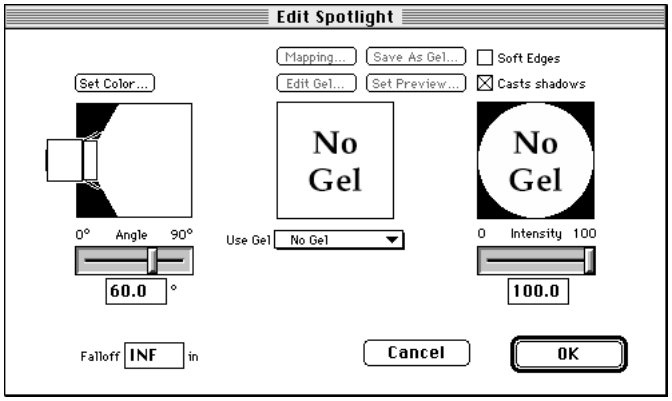
Another method is to use the Select Next command. Click on the spotlight icon. The sphere will be selected because its surface is the first encountered. Choose Select Next from the Selection sub-menu or use its keyboard equivalent, Command-tildé (⌘ ~) to ask Vision 3d to look deeper on the Z axis where you clicked. The spotlight should then become selected.

Whatever method of selection you use, remember that you must drag with the selection handles to keep the object selected when it is inside or behind another object. If, after selecting the second object, you click anywhere on that object's surface instead of a selection handle to drag it, the first object will be selected again and it will be moved instead.

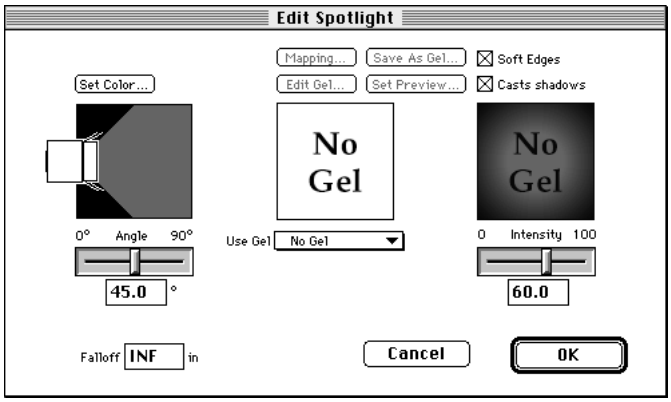
Once selected, there are several ways to move the spot lights. The most obvious is to select them in the Top view and drag them downward until they are approximately 1" to 1.5" from the sphere. A second way would be to use the Transform command and simply reposition them numerically an appropriate distance on the positive Z axis.

Now you will learn yet another way to move the spotlights. For this exercise, first switch back to the Front view if not already there. Select each spotlight in turn, then position the Object Move tool's cursor on either top corner handle and click-drag downward. Watch the feedback area at the top of the model window as you drag until the Z axis reads +3.500 (or as close to that location as the tool can position them). This pulls the spot lights toward the Front view plane on the Z axis. Refer to the previous exercise if you need help. The discussion in chapter 3 on dragging objects with their selection handles would also be useful.

5. **In the Front view, select the left spotlight.** This time the spot light will select as expected because it is outside of, or in front of, the sphere.
6. **Choose the Edit Primitive command in the Modeling menu.** Vision 3d opens the Edit Light dialog for the selected spotlight.

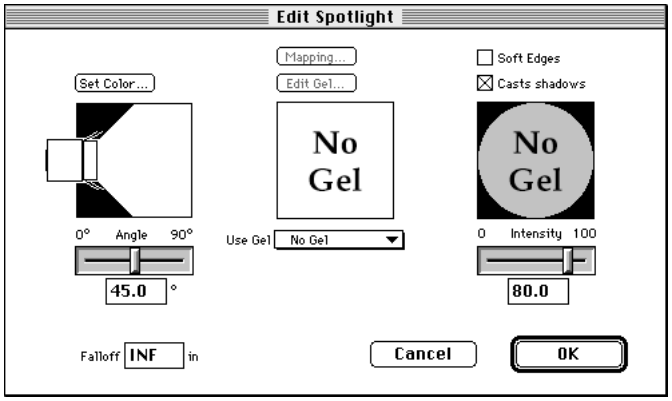


7. **Set the spot light's color to red.** Do this by clicking on the Set Color button. The Apple Color Picker appears. Choose a bright red color and click on the OK button. Vision 3d returns you to the Edit Spotlight dialog, where you can see the color you chose for the light represented.
8. **Further edit the light source's illumination parameters.** Do this by matching the settings shown in the following illustration, then click on the OK button. Vision 3d returns you to the model window.



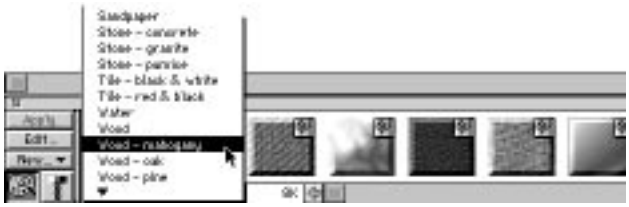
Change the Angle slider to 45 degrees. This control determines the spot light's cone angle — that is, how wide the field of illumination will be for the light. Enable the Soft Edges check box. Notice how the preview area in this dialog changes from a hard-edged shadow to a soft, feathered edge. Finally, adjust the intensity setting to 60 percent.

9. **Select the right spot light.** Open the Edit Light dialog. Again, this is done by choosing the Edit Primitive command in the Modeling menu. Alternately, you can double-click on a light source to open its Edit Light dialog box.
10. **Edit the spot light's illumination parameters as shown in the following illustration.** This time, set the Angle to 45° and the Intensity to 80%. Click on the OK button to return to the model window.



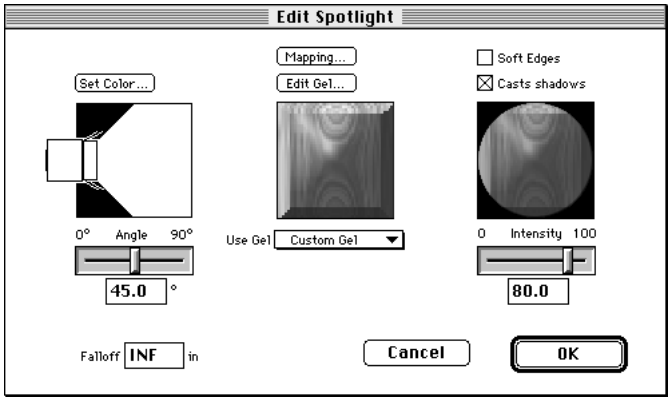
Note that you did not change this light source's color, use a gel, or enable soft edges. You will be applying a texture manually.

11. **Apply the Wood - mahogany texture to the spot light.** While the spot light is still selected and the Front view is the active view (so the texture will be applied from the back of the light source), use the Texture palette's pop-up menu to locate the Wood - mahogany texture. Selecting the Wood - mahogany texture from the pop-up menu causes the preview images to scroll on the palette until that texture is visible and a black rectangle appears around its preview to indicate selection.



With this texture selected on the Textures palette (a black rectangle is visible around its preview image), click on the Apply button. The selected spot light icon is visible in the model window and changes color to indicate the color of the texture that has been applied to it.

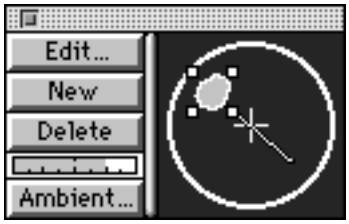
12. With the right spot light still selected, choose the Edit Primitive command. Vision 3d opens the Edit Light dialog box again for this spot light.



Of course, the previous cone angle and intensity settings are still in effect. But notice that you can see the Wood - mahogany texture that was applied in the model window now displayed as a custom gel in this dialog. Click on the Cancel button to exit this dialog. No changes need to be made.

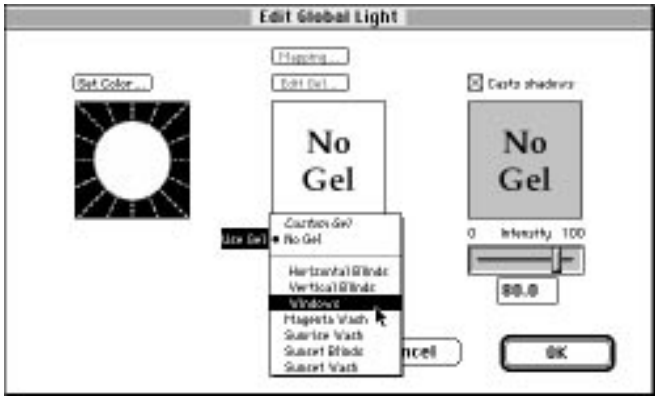
As discussed elsewhere in this chapter, applying a solid texture to a light source in the model window is the only way to use a solid texture as a light source gel.

13. **Next, select the default directional light source.** On the Lighting palette, click once on the head of the directional light source displayed there. Four selection handles should appear around the head.



14. **Click on the Edit button on the Lighting palette.** Vision 3d opens the Edit Light dialog box for the directional / global light sources. This looks similar to the Edit Light dialog for spot lights, but does not have the cone angle control, the Falloff field, or the Soft Edges checkbox.

15. **Click on the Use Gel pull-down menu and choose the "Windows" gel.** Vision 3d displays the Windows gel in the dialog.

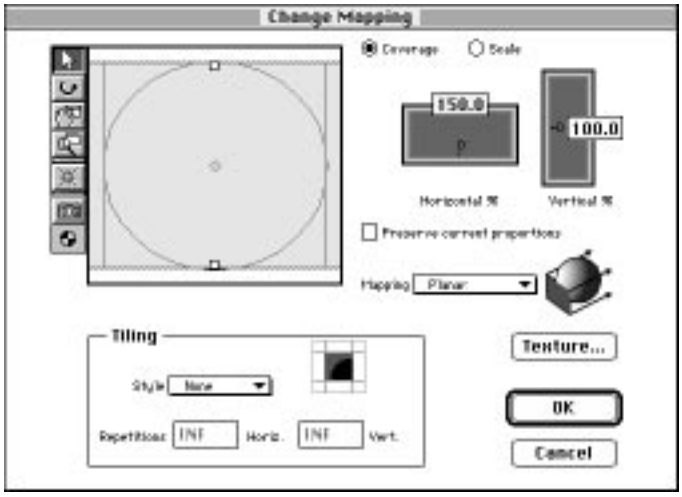


A gel is a special type of texture that is designed for use only on light sources. Gels do not appear in any other menu or list throughout the interface. Several gels are included to provide you with pre-configured special lighting effects. Of course, you can make your own gels to achieve custom lighting effects. Gel files can include color as well as shadow effects, but most are simple grayscale maps, like the Windows gel listed here in the Use Gel menu in the Edit Light dialog.

16. **Click on the Mapping button at the top of the Edit Light dialog.** Vision 3d opens the Change Mapping dialog. This is the same dialog used to edit regular texture maps applied to geometric objects in your model. However, it is used here to edit the gel applied to the directional light source.

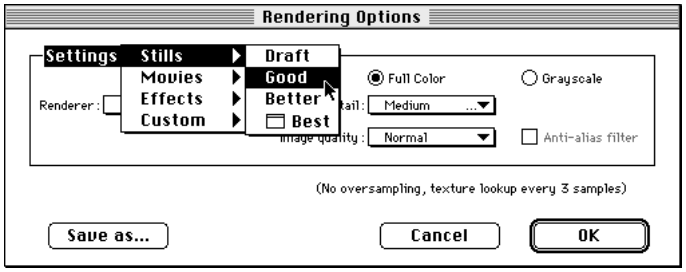
    NOTE: *The Camera tool in the Change Mapping dialog is not available when working with a light source. As discussed in chapter 5, this tool is provided to preview how a texture will appear on the selected object so you can make interactive adjustments while in the dialog. It would not be valid to render a texture on a light source, since the texture is intended to be projected FROM the light source, not to be visible ON its surface.*

17. **Match the settings shown in the following illustration.** Set Tiling Style to None. Set Mapping to Planar. Set Horizontal % to 150 and Vertical % to 100.



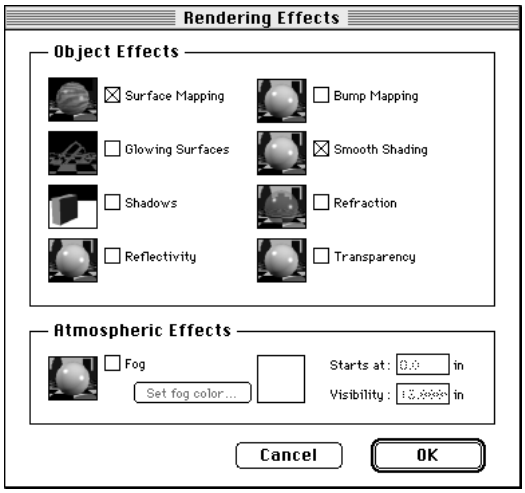
When the settings are in place, click on the OK button to get back to the Edit Global Light dialog window, then click on that dialog's OK button to return to the model window.

19. **Set up the rendering parameters.** It's time to see the results of your lighting efforts. Because the display in a modeling window is not of sufficient quality to see the lighting effects you have created, it is necessary to render an image of the sphere. Of course, you have to tell Vision 3d how realistic you want the rendered image to be. Since this is just a preview image, you don't want to waste time rendering a better image than you need. Typically, you will render two or three preview images, fine-tuning the details as you identify problems, then invest time on a final image with high quality options.



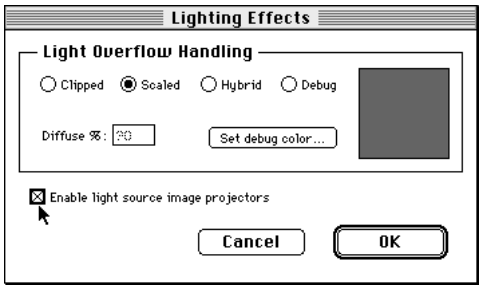
Pull down the Rendering menu and choose the Rendering Options command. Vision 3d opens the Rendering Options dialog window. From the Settings pull-down menu choose "Stills • Good" as your rendering setting. Then click on the OK button. Vision 3d returns you to the model window.

21. **Select your Rendering Effects.** Pull down the Rendering menu and choose the Rendering Effects command. Vision 3d opens the Rendering Effects dialog box. Match the settings shown in the following illustration, then click on the OK button. Vision 3d returns you to the model window.



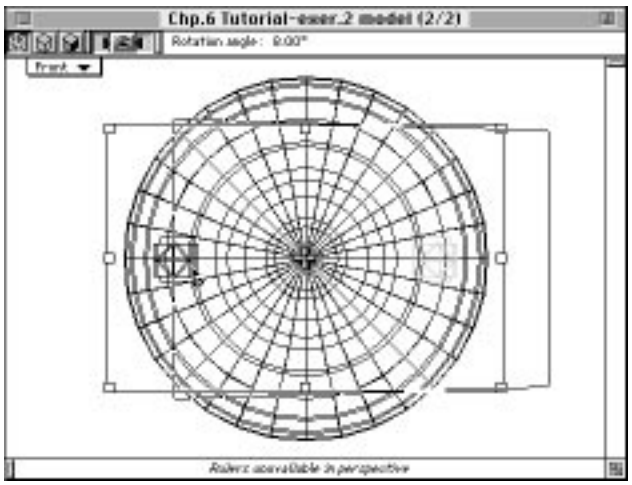
This dialog is where you decide which effects you want to see in the rendered image. In this case, only the Surface Mapping and Smooth Shading effects are enabled. There is no need to ask for any of the additional effects, since none of them will appear in the rendering.

22. **Set the Lighting Effects.** Pull down the Rendering menu and choose the Lighting Effects command. Vision 3d opens the Lighting Effects dialog box. Click on the "Scaled" radio button, and make sure the "Enable Light Sources as Image Projectors" checkbox has an "X" in it, so it is enabled. Then click on the OK button to return to the model window.



When you have multiple high-intensity light sources shining on one object, such as you do in this model, the scaled over-exposure control method does the best job. It is not as fast as the clipped method, but it is much more accurate.
Also, if the checkbox that allows light sources to project images is turned off, none of the effects you have set-up in this model would appear during rendering. Normally, this checkbox should be left on at all times.

23. **Set the view position.** The final step in preparing for a rendering is to pick a viewing position from which to render the image.



First, set the view to the Front view, if it is not already chosen. Next, switch the lens to Normal perspective. Use the “Fit Views

to All" command again to redraw the view at a scale that displays the sphere to best size. Finally, rotate the view slightly to the right so you can better see the wood grain projected by the right spot light. Do this by choosing the View Rotate tool on the Tool palette, then dragging the left center handle on the view management overlay 8° to the right (watch the feedback area at the top of the window as you rotate the view).

24. **Render your model.** If not selected already, select the large camera icon from the tool palette. Click once in the model window. Vision 3d begins rendering the model.

The finished rendering should look similar to the example shown here, with a spot of red light from the left spotlight with a soft edge, a wood grain pattern projected from the right spotlight with a hard edge, and the window frame pattern projected from the directional light source…all on the same object.

This second exercise has introduced you to gels and their effects on lighting a model. Of course, the rendered image is nothing special. It probably doesn't look like anything you would create yourself. But it does demonstrate the creative possibilities when using multiple light sources in a model. Hopefully, you can use the principles learned in these exercises, along with some experimentation of your own, to achieve nearly any conceivable lighting effect for your own projects.

Commands used in lighting control

The following commands are considered germane to the management of lighting functions in the software. A detailed explanation is included with each command listed.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in chapter 1.

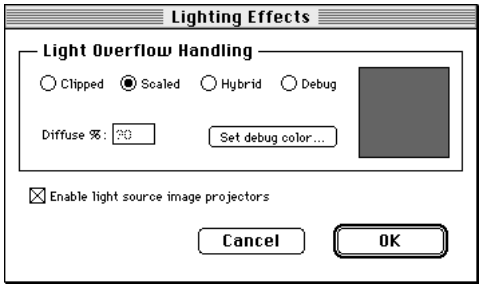
Edit Primitive command (Modeling menu, ⌘B)

Use this command to edit the illumination parameters of a light source. While this command provides some type of editing controls for each type of object, it is especially important for light sources. This command is the primary mechanism provided for adjusting color, light intensity, and many other light source settings.

Each of the three types of light sources has a slightly different set of controls when the Edit Primitive dialog opens, as is the case with all objects. These differences are discussed in detail at the beginning of this chapter.

Lighting Effects command (Rendering menu, ⌘J)

As well as enabling or disabling the use of light sources as image projectors, the Lighting Effects command is used to set the method of handling lighting overflow during the image rendering process. The default overflow handler is Clipped; the default for using light sources as image projects is enabled. If you want different settings, you can change them in this dialog, but your choice is not remembered between sessions.



An explanation of the fields contained in the Lighting Effects dialog follows.

Enable Projectors

Click this option to allow projection of gels or other textures applied to light sources. Clearing this checkbox turns off the projection feature globally. Depending on the nature of the model, disabling it may permit Vision 3d to execute faster renderings.

Lighting Overflow Handling

Automatic exposure control is a built-in feature of Vision 3d. The fields in this section of the Lighting Effects dialog box determine how excess lighting intensity in a model is resolved during the rendering process.

It is possible to over-illuminate an area of the rendering, just like in the real world. This may occur for different reasons:

- **By directing multiple light sources on certain objects.**
- **By setting the ambient lighting brightness too high.**
- **By giving object(s) surfaces high reflectivity, diffuse and ambient fractions.**

The result is a "white out" of the image, similar to an over-exposed photograph. Vision 3d has four ways to deal with light overflow problems. Only one of the four choices may be active at one time, with the following result:

- **Clipped:** Ignores all lighting intensity over 100%. This is the fastest method of overflow handling. However, it may cause color shifts in the rendered image. The quantity of light falling on a specific area is based on the total of the red, blue and green components of the light. Excess light is clipped as individual components, then recombined, which tends to affect each of the color components differently. If exact color is critical, clipping the excess light off is not the best choice.
- **Scaled:** This method scales total lighting intensity down so highest intensity is equal to 100%. It does not adjust the lighting if no area in the scene exceeds 100%. Scaled is the least likely to cause color shift problems, but takes a bit longer to render.
- **Hybrid**: If total lighting intensity exceeds 100% in a specific area of the model, this method divides the light striking the object(s) surfaces into its diffuse and specular components. It then scales back the diffuse light based on the value in the "Diffuse %" field,

and recombines the diffuse and specular components into a recalculated light level. Again, color shifts can occur during re-mixing of the lighting components, but this method gives you a greater degree of control in the final product than with the clipped method.

- **Debug:** If any lighting exceeds 100%, this option will show the overflow areas in the chosen debug color when the scene is rendered. You may then adjust the lighting components manually to reduce these "hot spots." To select the color, click on the set debug color button to display the color palette dialog.

Show/Hide Lighting Palette command (Windows menu)

The Lighting palette is always showing whenever Vision 3d is launched. You may have a need to hide or re-show the Lighting palette during the course of a Vision 3d work session.

The Show Lighting Palette command displays the Lighting palette. Show Lighting Palette is available when the Lighting palette is hidden. Hide Lighting Palette is available when the Lighting palette is showing. These two commands toggle in the same location on the Palettes sub-menu of the Windows menu.

Choosing the Hide Lighting Palette command is the same as clicking on the close box of the Lighting palette. Alternately, you can also show or hide this palette by clicking on the Lighting palette button on the Button bar. Refer to *Chapter 1: General Operation* for the location of this button.

CHAPTER 7:
SCRIPTING ANIMATIONS

Several textbooks have been written on the techniques of creating conventional animations. Those concepts apply equally well to computer-based 3-D models, though by their nature computer-generated animations are usually assembled from individual frames, rather than using the more traditional cell-based method.

Animation is the perception of motion over time, made possible by the marvelous way our brains process images transmitted from our eyes. The effect is called "persistence of vision." In animations, the perception of motion is achieved by presenting a series of finite images in rapid succession where the objects in motion are offset a small amount from one frame to the next. We see the motion because a persistence or residue of the previous image is still fading from our minds as each new image in the sequence is presented.

Vision's method for generating a series of frames to make an animation sequence starts with "event markers" placed at intervals along a time line. Based on the spacing of the event markers and the number of frames per second used to create the movement, Vision 3d calculates how each in-between frame should render to achieve a fluid motion to the movement of objects within the model. Calculating the in-between frames is called "tweening." The more frames per second, the smoother the animation…and the longer it takes to create.

An important point to remember here is that the eye is very forgiving of lack of detail while an object is in motion. While a single frame of the model might require the full suite of rendering effects to produce an interesting still image, an animation sequence can often be rendered with fewer effects, and perhaps in one of the lesser-quality rendering methods, with excellent results.

***NOTE**: An example is presented at the first of this chapter to illustrate the effective use of hierarchical animation. It demonstrates recommended techniques for scripting complex motion, using the elementary sun, earth and moon orbital mechanics. It is also used as a vehicle to explain the operation of the Animation palette. You will want to read and follow along with that example to get the most from this chapter. Other tutorials and a control-by-control explanation of the*

Animation palette are included later in the chapter for review, but this is the most comprehensive description.

Hardware limitations

When creating an animation in Vision 3d, the final use must be kept in mind. The final use usually determines the file format and compression method for the rendered animation frames. Experience is the best teacher of what works best in each situation. If you are working with a service bureau for outputting your final product, they may have some specific recommendations and requirements for the material you provide them.

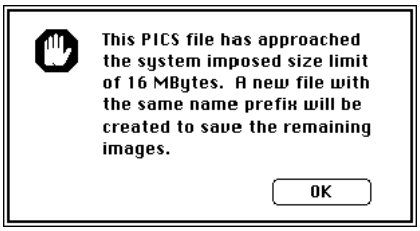
Some projects require short sequences and small frame sizes for use in interactive on-screen projects. In that case, you want to keep the animation in digital form to put the finishing touches on your project with multi-media software. Others projects may be intended for merging into a finished video sequence for commercial use. You may wish to output the rendered animation in raw form that can be edited and / or mixed conventionally with other video and sound to arrive at a finished product. Vision 3d can produce animation files appropriate to the venue in which you choose to work.

Commercially-viable animation is fairly recent to the Macintosh. There are some tools in the desktop animation area that still must mature further. Theoretically, there is no upper limit to the number of frames per second or the finished length of an animation. There are some very real physical limitations, however, related to the current state of the Macintosh hardware and released system software and extensions. Some of the areas that may impact your creativity are as follows:

- **PICS Files cannot contain more than 16 megabytes.** The Mac OS does not support files with more than 16 Mbytes in the resource fork. That is the way the PICS format works — all images are stored in the resource fork. But, saving a file with more than 16 megabytes of resources to disk can be disastrous to the data in that file as well as existing files on the same volume. When MacroMind Inc. (now known as MacroMedia Interactive Inc.) devised this file format for use with their Director™ application, there weren't the alternatives that are available now, i.e. — those provided through QuickTime, so it was quickly adopted by other applications to "connect" more easily in the fledgling animation / multi-media arena. The truth remains: The

resource fork was never intended to perform these duties meant for the data fork. Use of the PICS format is lessening as more and more applications become QuickTime-aware and the tools become more mature.

PICS files containing multiple PICT images of an animation can easily exceed the 16 Mbyte size, because ALL the frames of an animation are stored in the resource fork. Given this limitation, Vision 3d will not assemble a single animation sequence in PICS format larger than 16 megabytes.


This PICS file has approached the system imposed size limit of 16 MBytes. A new file with the same name prefix will be created to save the remaining images.
OK

The obvious work around is to save long PICS-formatted animations in segments of less than 16 megabytes. You would have a difficult time determining that yourself. So, in the process of rendering and saving the frames of an animation, if Vision 3d detects that a PICS-formated animation will exceed the 16-megabyte limit with the next frame, a directory dialog appears asking for the name of a second file.

- **QuickTime Movie files can spool faster from the hard disk because they are compressed.** There are several compression methods available in QuickTime; all of them use a “lossy” format. That is, they discard data from the images to achieve the compression. There is a trade off between the picture quality and compression ratio, which is generally inversely proportional. A high quality setting will achieve low compression. Alternately, image quality is usually sacrificed to achieve high compression. While different compression schemes may produce varying incremental changes, the general rule is that quality goes up as compression ratio (and thus playback performance) goes down, and vice versa.
- **The maximum number of frames per second that can be played is highly dependent on the frame size, the color depth, and the hardware architecture.** Obviously, a sequence of smaller one-bit

(B&W) images will spool from disk onto the screen faster than a set of larger 32-bit (full color) images.

CONSIDER: *A simple stream of moving color images can quickly become a torrent of bytes that brings most Macintosh hardware to its knees. For example, to play an animation made of 32-bit, 72 d.p.i. color images the size of a 13 inch screen at 30 frames per second requires data to pour in from the hard disk at a rate of 20 to 36 megabytes per second, depending on the compression method used. Even the lower end of this range is more than can be handled by even the fastest Macintosh hardware. It is faster than can be handled by the standard SCSI connection; It is faster than Color QuickDraw can display it on the screen; It is even faster than the data can be transferred to video cards through the NuBus architecture. And if you want to add sound to the animation, the discrepancy in capacity gets even more absurd. Full-screen, full-color, full-motion animation on a standard Macintosh is not possible in real time without specific hardware support.*

Obviously, faster platforms will be built and quicker transfer rates will be provided to better support the development of true multimedia production. There are some hardware alternatives available now — such as CPU accelerators, real-time data compression / expansion cards, QuickDraw video accelerators, faster storage / retrieval devices, SCSI-2 ports (with both Fast and Wide capabilities), separate I / O handlers, etc.

This discussion about limitations is presented to give you a more realistic expectation about what you can do in your "cinematic debut." Vision's role in producing the finished animation is to assemble 3-D models and render raw footage of them in motion or with various animation effects. No tools are provided to edit the animation once it is rendered. Those are available in other applications intended for that purpose.

While digital animation-editing tools are improving all the time, professionals usually prefer to output an animation sequence one frame at a time to video tape, then edit the finished animation on video. This requires a "frame accurate" video controller and compatible tape deck. While still relatively expensive, they have become more affordable in recent years.

Do not expect to become the electronic equivalent of John Ford overnight. That will take some work and many projects to hone your

directorial skills. Besides, anything you can do on a Macintosh has probably been done better by George Lucas and his company, Industrial Light & Magic, Inc. (of the Star Wars trilogy and many subsequent films using incredible special effects) using high end computers, specialized equipment, and LOTS of people with multi-million dollar production budgets. That said, you may be the first to use your ingenuity and prove us wrong!

Yes, you can create broadcast-quality animations on the Macintosh. You will be able to amaze your friends and you will have lots of fun (along with some frustrations) in the process. Understand that it takes more software than Vision 3d to produce finished animations (with sound, special effects, A-B rolls, etc.). And even though animations can now be done entirely on computer, meaningful results still require skills in the traditional areas of story boarding, cinematography, and scene direction. *But if it was easy, everyone could do it and you couldn't make any money!*

Concept of "actions" in creating motion

Complex motion does not happen by accident. It must be planned and deliberately scripted. Executed ineffectively, at best the result is often just a busy screen. At worst, the objects are not well controlled and the project is usually abandoned before completion because "it can't be done." Done well, however, the animation tells your story and you are commercially rewarded.

Ironically, the best tool you have for creating award-winning computer animations is not on the computer…it's paper and pencil. Investing the necessary time BEFORE you begin a project to write down (or draw) a story board and work out all the details, including special notations and time indexes, can be your greatest help to keep focused and on-task when you are in the thick of a project.

Actions that are time-dependent

In animation terminology, "action" refers to motion. The motion may be either with the objects, the camera, or both. An action is a single movement over a specific sequence of frames. Vision 3d measures the following actions as it interpolates, or 'tweens frames for an animation sequence:

- **Position.** An object may move from one location in 3-D space to another over time. Directional and local light sources may be moved the same way as geometric objects. The speed of

movement is determined by the distance and the time given for it to move. Note that the path or action is always a spline "path" in Vision 3d.

- **Rotation.** An object may rotate on any axis or combination of axes over time. The speed of the rotation is determined by the angle specified between event markers and the time given for the rotation. The rotation action may be linear or spline (while a little harder to conceptualize, the same tweening parameters applicable to position also apply to rotation).
- **Scale.** An object may be scaled larger or smaller over time. The speed of a scale change is determined by the distance and the time given for it to occur. The action may be linear or spline.
- **Texture.** An object or light source may change its texture over time. It may change as often as desired. The timing of the transition(s) are controlled by the placement of event marker and their type (linear or spline).
- **Light intensity.** Light source objects may change intensity over time. This includes directional / global lights as well as the local light types.
- **Environment.** The environment may be changed over time by using the Rotoscoper solid texture extension to "play" an animation as the image shown or reflected. While both the background and reflected environments are aware of time, note that you cannot simply change textures over time as you can with an object or light source. You must use the Rotoscoper extension. Please refer to Chapter 5: Applying Textures for more information on the Rotoscoper extension.

Multiple object and camera actions may occur simultaneously during the same frame sequences. You may use as many of these actions as needed in the sequence to create the animation effects you desire.

The following parameters are used in an animation, but not 'tweened, i.e. — they are manually set for the entire sequence:

- **Frame size of rendered animation.**
- **Focal length of lens.**
- **Viewing position (in model windows).**
- **Object geometry.**

From the first list, you can see that Vision 3d is able to monitor quite a few actions in its tweening calculations to produce full object

and / or camera choreography. However, no time dependencies are attached to the items in the second list. They are considered static and must be set before each rendering session.

Minimum requirements for animation

The following points are presented as suggestions for how to accomplish some of the basic effects available in Vision 3d:

- **A minimum of two event markers at different points along the time line are required for animation to occur.** There is no limit to the number of event markers that may be defined, depending on the available memory. During preview playbacks or the high-level rendering of each frame in an animation, Vision 3d then moves each animated object from one event marker to the next, yielding as many positions between the event markers as requested by the frames-per-second setting.
- **Rendered animation sequences may be generated from any view.** The active view determines the viewing position used for the rendered frames of an animation. You may use a view in a modeling window or the view from a camera window. In the case of a modeling window, no view movement is possible, since the viewing position in a modeling window cannot be moved over time. It is fixed. Camera objects are provided for this purpose.

  To add camera motion to an animation, first insert at least one camera object in the model and give it an animation path just like any other object. Then you can use the camera window attached to that camera object as the source of your animation's images. Since the camera window shows the view of a model from the selected camera object, the viewing position of the animation can change as the camera moves. You may wish to do the same animation sequence from several different viewing positions — from static views and from moving cameras — then combine the rendered sequences in an animation-editing application for the finished "movie."
- **Life of object.** If an object is added to the model while the time pointer is at a position other than time zero on the time line, that object is assumed to have existed in that same position from time zero. This is the default for each object.

  A method for making objects appear and / or disappear is to create and apply a fully-transparent texture to an object. The

texture applied to an object may be changed over time. Thus, it would be possible to hide an object by making it completely transparent, then change to another texture at the appropriate time that would cause the object to become visible. This is useful if you need the object to be moving at the moment of birth.

- **Editing textures during surface metamorphosis.** Textures may be changed on an object over time. During the transition from one texture to the next, there will be some frames in which the object will actually have two textures applied to it at the same time. This can be a useful technique, even for still images. But, it also means that the "Edit Selected" command in the Textures menu will not work for the object while in metamorphosis. Basically, it does not know which texture to edit. When that situation exists on the time, an alert will appear.

  You can, of course, still edit any of the individual textures involved in a metamorphosis through the standard texture-editing commands, including those found in the Textures browser. All Textures are considered to exist in their current definition from time zero. Their use may be time dependent, but the texture definitions themselves are not.

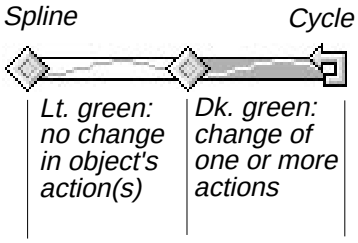
Event marker description

Vision 3d calculates the motion of objects by the control of their event markers as they follow a motion path. The setting for all event markers is to construct a spline path, with smooth transitions between each marker.

NOTE: *Motion paths are obvious and easy to understand when an object moves through 3-D space. A line can be seen showing exactly where and how the object is moving. The concept of a "path" is just as valid for actions such as rotation, scale, texture morphing, focal length, light intensity, etc., even though they do not leave a visible path in the model windows.*

Two types of event markers

...and variations on their path control



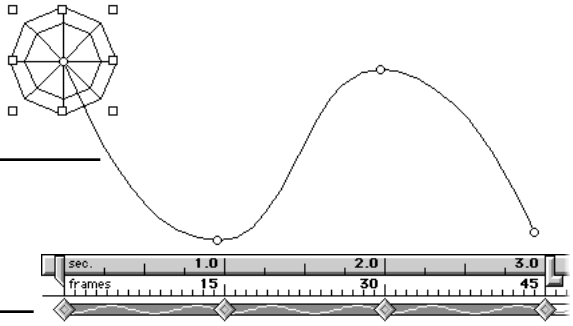
The color between event markers is determined by the presence or lack of object action(s) between markers. The above illustration identifies time-dependent changes in an object by a dark green color. If there is no "tween-able" changes in an object from one marker to the next, a light green color appears. Thus, depending on the action scripted, the behavior of the object may be different. (On grayscale monitors, the colors become dark gray and light gray. Monochrome monitors show black and white. The wavy line is inverted in the black so it can still be seen.)

Visible motion paths

Whenever an object is selected that has had an animation scripted for it AND the Show Animation Paths command is enabled in the Animation menu, the path for that object becomes visible in the modeling views.

The paths of animated objects are visible in the model windows when selected

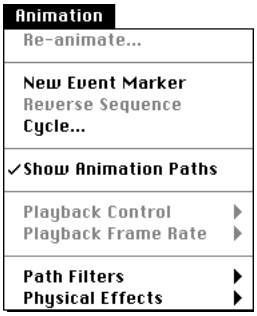
Points on the visible paths correspond to the event markers in the Animation palette



The animation path appears with points on it that correspond to event markers on the Animation palette. Just as you can click and drag an event marker to a different moment on the time line, you can also drag the path's points in 3-D space. Thus, where the event markers allow you to set the temporal relationship of an object's motion (the "when" of its motion), the points on a visible path allow you to set the spatial relationship of those same actions (the "where" of its motion).

Clicking on an event marker changes the color of the icon and its corresponding mark on the path. Likewise, clicking on a point on the motion path also selects the corresponding event marker on the animation palette. The area inside an icon turns red (dark gray on grayscale monitors, black on monochrome) to indicate selection.

To select more than one event marker at a time, you can drag a marquee around the desired markers or you can hold down the Shift key while clicking on them one at a time. The Shift key can also be used to select multiple points on a visible path. Note that you cannot use the marquee drag method for selecting multiple points, because dragging a selection marquee in a modeling view de-selects the previous object(s), including their motion paths.



If you do not wish to clutter the views with motion paths every time you select one or more objects, you can disable the Show Animation Paths command. The command toggles between on or off each time it is chosen in the menu.

Using hierarchical scripting

Individual event markers in Vision 3d can accept multiple actions for an object, such as changing position and rotating at the same time. This is often how animations are developed. However, it can sometimes be difficult to go back later to alter or change one of those actions without affecting other actions that may also be part of the same event markers. That is when animations can become unwieldy and difficult to control.

The solution is to identify the separate actions making up each object's motion, then script those actions in a structured manner. It would be much more preferable if you could script the individual actions separately, so that if the need arises you can edit them separately; confident that you will not impact other aspects of an animation that you do not wish to change. Vision 3d provides a mechanism for doing just that. It's the same Shapes functionality presented in Chapter 4, now applied to animations. If you have not

read that chapter, you should take time to do that now, before exploring Vision's animation capabilities further.

As you know from that chapter, to use the shape feature effectively requires some pre-project planning. Once you take the time to put your animation on paper, it is usually a simple matter to break down the motion for each object into a series of simple actions. Do you remember the discussion in Chapter 4 about the ability to model complex geometry using shapes and instances of shapes? And how you could build complex objects and/or scenes by assembling from simpler objects that could be worked on in their own isolated workspace? That same concept can be also applied to animation actions. You can use various shape levels to script simple actions; adding them together at a higher level to form the desired complex motion.

An example: Scripting simple actions at different hierarchical levels

Suppose you want to create an animation illustrating the orbital relationships between the sun, earth, and moon. The motion of the moon must work so that as it rotates on its own axis, it also rotates around the earth AND moves with the earth as it rotates around the sun. Even such a simple example as this can be rather intimidating if you try to do it all in the model space. How can you maintain the orbit of the moon around the earth when the earth is moving in its own orbit as the same time? Obviously, using different levels of shape workspaces to script the rotation of the moon and earth can make the project much simpler.

To keep things simple, this project will not address the application of textures. After the model is done and a first rendering of the animation is completed, you can go back and apply textures for a more complete look. For the moon and earth, this would be done in their root shapes. The sun is just a sphere in the model, so that should be simple enough.

One more point. Since the actions you will script in this exercise are essentially all rotations about an axis, there will not be any visible animation paths. As discussed earlier, the paths are there, but since the origin points about which the objects rotate or orbit do not move the paths do not show. The exercise is still valid, and you can still apply the concepts to your own projects where an object's motion will show a path.

The planning

At the beginning, you must decide the size of the objects. This is required so you can then decide the diameter of their orbits. In this example, the sun will be ten inches in diameter, the earth will be three inches in diameter, and the moon will be one inch in diameter. Further, the earth's orbit will be scripted at 50 inches and the moon's orbit set to 15 inches in diameter. Obviously, these objects and orbits are not to scale, but then what example of the Solar System ever is? It would be possible in Vision 3d to build this project to scale, since there is no boundary to its 3-D space. You might have some concern with the software's math precision at that scale, however. And you definitely would have trouble seeing the earth and moon from any comprehensive viewing position and scale setting.

Next, you must decide how long to make the animation so you can set the rotational and orbital periods to complete within that time frame. It would be ideal if you could render the animation so that it loops without a seam. That would allow you to play it continuously; giving the effect of a very long animation.

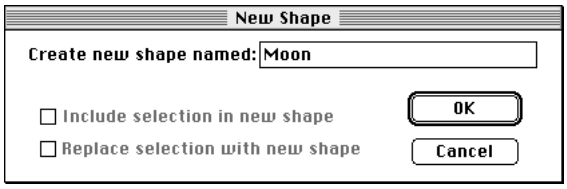
In this example, the animation will be nine seconds. During that interval, the earth will rotate once around the sun in that time (the equivalent of one year…talk about compressing time!). The moon will rotate six times around the earth or once every 1.5 seconds. Okay, so the moon actually orbits the earth a little more than thirteen times a year, but in this project, whipping the moon around the earth that many times in nine seconds will cause significant temporal aliasing (discontinuity or jumps in the motion). Slowing things down a bit will make a smoother, more pleasing, and more understandable animation. (Maybe you could script a version that is more accurate after you see how this is done.)

Although not necessary, you may also want to make a conscious decision to center all orbits at 0,0,0 in their respective workspaces, so you can accurately set the offset for the radii of all objects' orbits. It will make the animation scripting a little easier if you are consistent in this respect.

The Moon

Begin by creating a new model. Within the new model, create a new shape. (Use the New button on the Shapes palette.) Enter the name of this new shape as "Moon-rotating." This is a root shape. At

each instance (or level change) of the moon, you will add more action; building up the complex motion that the moon will be required to perform.



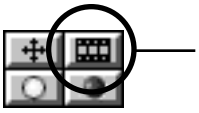
Within the Moon-rotating shape window, switch the view to the Top, then draw a sphere one inch in diameter. Position it at the center of the workspace (X=0, Y=0, Z=0). Notice that the sphere is rotated 90° so that the poles are oriented top and bottom when looked at from the Front view. Please refer to Chapter 3 if you need help doing this task. (Here's a hint: Use the Transform command in absolute mode. If you don't know how to do something as simple as positioning an object in a work space, perhaps this is not the chapter you should be reading right now!)

So far so good. This root shape level is where you will give the moon rotation on its own axis. From your Junior High science class, you remember that the moon rotates on its axis in exactly the same period as it takes to orbit the earth. (That's why it was not until the 1960s that humans were able to get pictures of the back side of the moon to see what it looked like, because only one side is ever visible from the earth!)

To script the moon's rotation:

1. **Show the Animation palette, if not already present on the desktop.** This can be done through the Palettes sub-menu in the Windows menu. Alternately, you can click on the Animation palette's show / hide button on the Button bar.

Clicking on the Animation palette button in the Button bar toggles it to show or hide



Animation palette's show/hide button

The show or hide status of the Animation palette is retained from session to session. If it was showing when you quit

Vision 3d the last time, it will be showing at the start of the next session, and vice versa of course.

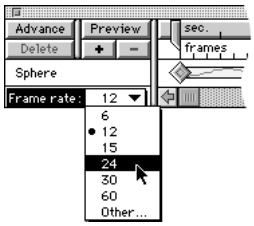
2. **Click and drag the Animation Length Pointer to nine seconds.** The palette automatically scrolls horizontally as you drag the length pointer out to the nine second mark.

To set the animation length for the document, drag the L-shaped icon

Line up the vertical line at the left edge of the pointer on the time mark that you want

This sets the animation length for the entire document.

3. **Set the frame rate to 24 frames per second.** Click on the Frame pull-down menu on the left side of the palette and choose 24 from the list of options. The default frame rate for a new model is 12 frames per second — usable for previewing, but not smooth enough for finished work, or situations like this project where a lot of motion is packed into a short duration.

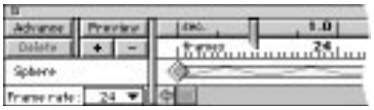


Notice how the frame count changes along the time line on the palette to correspond to this new setting. For nine seconds of animation, you are now working with 216 frames that must be rendered to create the animation images. Of course, the higher the frame rate, the smoother the motion.

NOTE: *If you were intending to eventually output this animation to video tape, you would need to set the frame rate to 30 frames per second, or 270 total frames for the nine seconds specified in this project.*

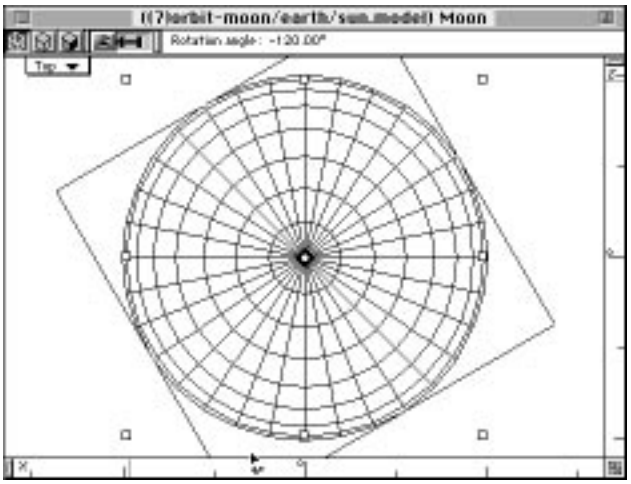
4. **Select the Object Rotate tool from the Tool palette.** Please refer to chapters 1 and 3 concerning the location and operation of the Object Rotate tool.

5. **Move the current time pointer to the half-second mark on the time line.** The time pointer will snap to each major tick mark on the time line as it gets close. You can check the time field, which indicates the pointer's position as you drag and release.



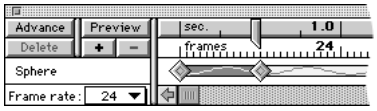
Alternately, you can click on the Time button to enter the desired current time directly. The time pointer moves to the value set in the dialog when you click on the OK button. There is also an Advance button on the top left of the palette. Clicking on this button moves the current time pointer to the next whole second on the time line, which is too far for this step but may be useful in other situations.

6. **Rotate the sphere 120° counter-clockwise in the Top view.** Do this by positioning the Object Move tool's cursor over the top left corner selection handle of the sphere. Click and drag downward until the object is rotated 120°. Watch the feedback area at the top of the window as you drag.



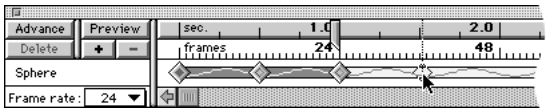
Releasing the mouse button executes the rotation and causes the view to redraw. It also causes an event marker to be

automatically inserted on the Animation palette for the sphere object at the current time.



The area between the event markers at time zero and at 0.5 seconds changes to a dark green (on color monitors, of course) to indicate there is some action taking place. A light green color means there is no change in the object between adjacent event markers.

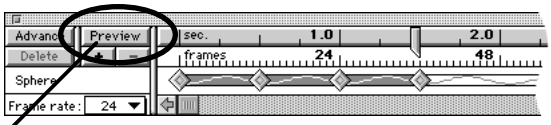
7. **Repeat the rotation another 120°.** This means moving the time pointer forward another half-second (to the full second mark), then rotating the sphere 120° again in the same direction. A third event marker appears on the palette.
8. **Duplicate the event marker at time zero and position it at 1.5 seconds.** This is done by holding down the Option key while dragging the time zero event marker. The original marker is left where it is and a copy is dragged to the new location.



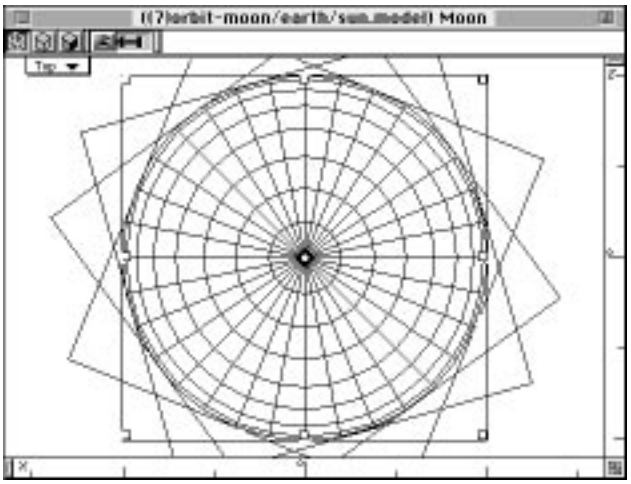
Releasing the mouse button drops the duplicate event marker onto the time line. Notice that you can drag event markers past the current time pointer, but not the Length pointer. Also notice that the time pointer does not automatically move to the time of the newly-inserted event marker.

Why use a duplicate of the first event marker? Yes, you could have simply rotated the sphere another 120° to complete the 360° rotation, and it probably would have worked just fine. However, by using a copy of the event marker at time zero, which is the position of the sphere BEFORE it starts rotating, you can guarantee that the sphere's rotation will return to EXACTLY the same location. That further guarantees that the rotation will be seamless, without a hitch or hesitation.

Clicking on Preview button starts a run through of the animation script using bounding boxes for the objects



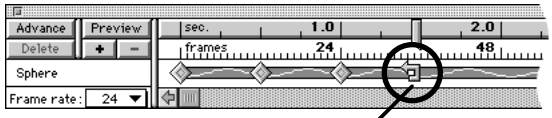
To double-check the rotation script, click on the Preview button at the top left of the palette. You should see a bounding box rotate around the sphere in the shape window.



It should rotate smoothly for one full 360° interval, then stop as the preview proceeds along to the nine second limit. Any time during a preview run through — in this case, wait until after it has passed the 1.5 seconds mark — you can press the Command-period (⌘.) key combination to stop it.

9. **Move the time pointer to 1.5 seconds, then choose the Cycle command in the Animation menu.** Click on the "Enable cycling…" checkbox and set the cycle back field to 0.00 seconds, if not already there, and click on the OK button.

The last event marker will change appearance to indicate cycling is enabled for this object. You can disable the cycle feature, as needed, by choosing the Cycle command again and turning off the checkbox.



Note how the event marker icon at which the cycle option is enabled changes appearance

Notice how the dark green "action line" now continues past the last event marker, providing further visual feedback of the cycle option in place for this object.

Cycling is a very useful feature that eliminates the need to manually script a repetitive action, such as rotation, for the full length of an animation. By laying down event markers for a single iteration of the action, then turning on cycling, the object will repeat the action *in perpetuity*, or to the end of the designated animation length (whichever comes first). If you make the animation length longer, there is no need to re-enable the cycling.

Clicking on the Preview button now shows that the sphere continues to rotate AFTER the last event marker, just as if you had scripted the rotation manually.

This concludes the work you need to do in this shape workspace. You can close the moon's shape window to make room for other windows. Of course, the shape is still part of the model, even if its window is not open on the desktop.

Now would be a good time to save your model if you have not already done so. As with any application, it is always a good idea to save often, especially after completing a major phase of the project.

The Earth

The next phase starts by setting up a new shape for the earth object. Title it, "Earth-rotating." This too will be a root shape where you will script the rotation of a sphere on its own axis. While some of the controls on the Animation palette are global and have effect for the entire document (such as the Animation Length pointer and the Frames per Second setting). Other controls are applicable only to the

current level of the document. Because you are now in a different workspace, the work you did in the other workspace or level is not displayed here. Be assured that the work is still in place, and would reappear on the palette if you were to open the Moon-rotating shape window and select it again.

Before you do anything in this shape workspace, move the current time pointer back to time zero. THIS IS IMPORTANT! If the time pointer is at another point on the time line when you draw the sphere, the object is automatically assumed to have existed in that state as it was drawn from time zero. Then as you adjust its size and position to begin the rotation scripting, event marker will be added for the object as you make change, but not at time zero where you want it to be. It is important that you begin factoring time into your modeling techniques or you will get little (and big) surprises that may be hard to reverse unless you understand what is happening to your model.

Draw a sphere in the earth's shape workspace. Be sure you are in the Top view when you do this, just as you did in the moon shape. As discussed at the beginning of the project, make this sphere three inches in diameter and position it at 0,0,0 in the workspace. Using EXACTLY the same steps outlined above to script the moon's rotation, i.e. — script a 360° rotation for the earth. The one difference here is that you want the earth to rotate once every second instead of 1.5 seconds. Thus, you must set your event markers for 120° rotation each at .3333 second, .6667 second, and one second, respectively. This corresponds to frame 8, frame 16, and frame 24 of the animation. Of course, the cycle should be enabled from one second back to zero. (This means the earth will only rotate nine times in a full orbit around the sun…again for animation clarity at the expense of accuracy.)

When you are satisfied everything is working as it should, i.e. — just like in the moon workspace only with a slightly faster rotation, close the window. Be sure to save the model now, before moving to the next part of the project.

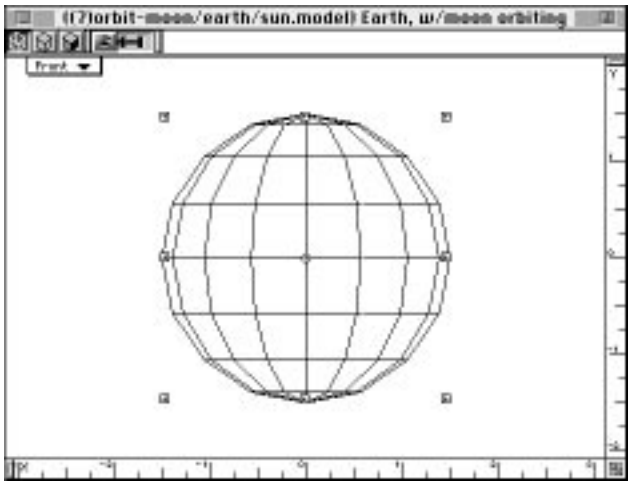
The moon's orbit

In this phase, you will bring the earth and moon together, then lay down event markers that describe the moon's orbit around the earth. This is a second action added to the moon's previously-scripted rotation. Combining the two actions can be hard if assembled wrong, but ridiculously easy when done correctly.

Again, make sure the time pointer is re-set back to time zero before inserting the earth and moon shape instances. This is mentioned again to help develop a habit. Each time you move to a different level within a model, you should move the time pointer to time zero before beginning any work. Otherwise you may not realize when you are changing something that you didn't intend.

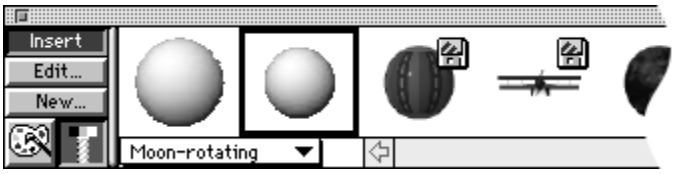
To script the moon's orbit around the earth:

1. **Begin by creating another new shape workspace.** Title it, "Earth w/moon orbiting." A new shape window opens. As before, be sure to return the time pointer to time zero BEFORE putting anything into this workspace.
2. **Insert the earth into this shape's workspace.** This is done by choosing the earth shape on the Shapes palette and clicking on the Insert command. This inserts a copy of the earth shape at the same size. Just to make sure, use the Transform command to fine-tune the object's position and size. Notice that the earth object is inserted in the same orientation as it was in its own workspace, even though you are in the Front view.



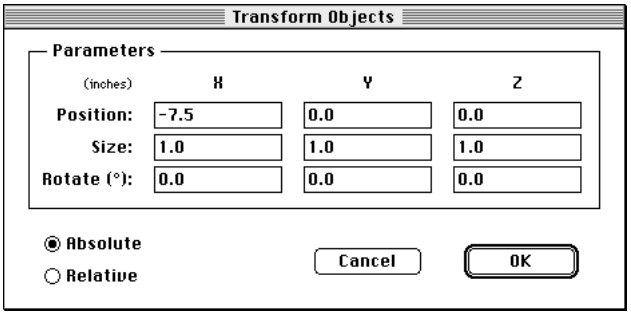
More accurately, you are inserting an instance of the "Earth-rotating" shape into a new shape. The instance carries all of the animation scripting entered in the original shape to this level. No event markers for the previous level's animation scripting appear here, but if you click on the Preview button, you will see the rotation action working on the instance.

3. **Insert the moon.** Do this in the same manner. With the palette switched to the Shapes palette, then select and insert the "Moon-rotating" shape using the controls and buttons on the Shapes palette.



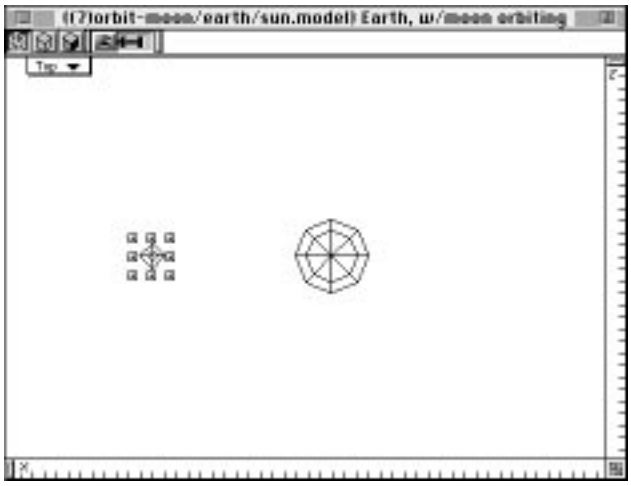
As before, this has the benefit of automatically inserting the shape at the view set center (which should be at 0,0,0 by default) and at the same size.

4. **Move the moon 7.5 inches away from the earth.** As discussed in the introduction to this project, the moon's orbiting diameter is 15 inches, so its offset from the earth would be 7.5 inches as the radius.



The easiest way to do this is to use the Transform dialog. While the moon is still selected (as it sits inside the earth, since both were inserted at 0,0,0), choose the Transform command and enter -7.5 in the X axis field. Clicking on the OK button shifts the moon 7.5 inches to the left on the X or horizontal axis, as observed from the Front view.

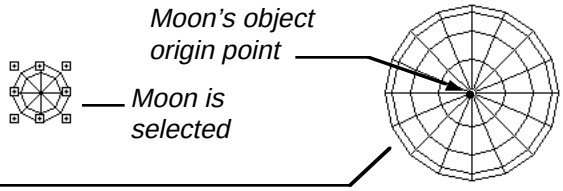
5. **Switch to the Top view and zoom out until you can see both objects.** Obviously, the orbit will be on the X-axis' plane, so the Top view will give the best vantage point from which to script the orbiting action.



Using the Magnify tool, hold down the Option key while clicking in the active view. Each click zooms the view's scale out by a 2X factor. You will need to click twice to see the moon in the view. This is done so you will have enough room in the view to rotate the moon around earth as you lay down the event markers.

6. **Move the moon's object origin point to the center of the earth.** The object origin point is the dot that appears along with the selection handles for an object.

Dragging the moon's origin point to the earth with the Control key held down offsets the moon's origin point to the center of the earth



Moving the origin point for the moon requires the use of a modifier key while dragging the dot. Otherwise, the entire object moves. Hold down the Control key, then click and drag the origin point horizontally with the Object Move tool. As you are dragging, you can also hold down the Shift key to constrain the direction the origin point moves. This insures that it is moved EXACTLY horizontally to the center of the earth. Now, whenever

the moon is selected in this shape workspace, its origin point will appear in the center of the earth.

7. **Move the time pointer to .5 seconds.** As discussed, the moon's orbit has the same period as its rotation. Thus, you will lay down three event markers that describe an orbit duration of 1.5 seconds.

8. **Rotate the moon around the earth in 120° increments.** Using the Object Rotate tool, with the moon object selected, click and drag a corner handle downward so that the orbit is counter-clockwise, just like the moon's rotation. As before, releasing the mouse button at 120° rotation causes the screen to redraw and a new event marker to appear on the Animation palette.

   Notice how the moon rotates around its origin point, which is at the center of the earth, making it very easy to describe a nice circular orbit centered on the earth.

   Next, advance the time pointer to one second and rotate the moon another 120°. Then, option-drag the moon's first event marker to the 1.5 second mark. And finally, click on the last event marker (to select it), then enable cycling back to time zero for the object. Notice that the time pointer did not have to be moved to the 1.5 second position in order to use the Cycle command.

   Click on the Preview button to check the orbiting action. Notice how the moon also rotates as it orbits the earth, which is also rotating as prescribed. Notice also how the motion continues past the event markers you have laid down on the time line.

This concludes the third phase of the project. When you are satisfied that the animation scripting is correct in this shape, close the window. Remember to save.

THE EARTH'S ORBIT

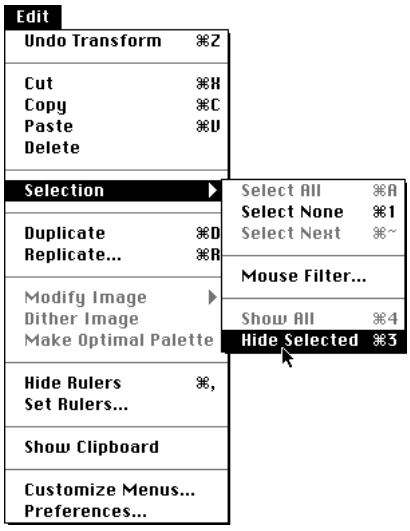
In this last phase, you will script the animation of the earth's orbit around the sun. This is where you will finally insert objects into the main model space. You will draw a large sphere for the sun, then insert the "Earth w/moon rotating" shape and give that instance an orbital action around the sun.

Yes, it is typical for a lot of work to be done in shape workspaces before beginning the assembly process in the main model space…at least for projects that are well-structured. Vision 3d is very flexible; it allows you to build models any way you want. This project is an

example of the recommended way to build models…the way that takes advantage of the program's modeling strengths.

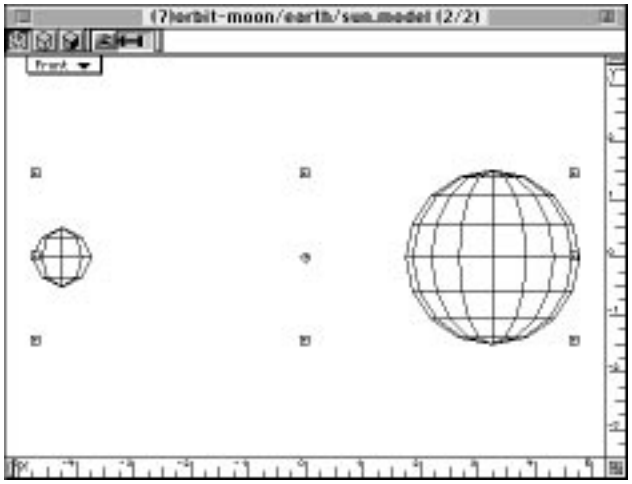
To script the earth's orbit around the sun:

1. **Begin this phase by switching the model window's view to the Top view.** You are preparing to draw the sphere for the sun in an orientation consistent with the other objects. Again, make sure you move the time pointer back to time zero before beginning any work in this workspace.
2. **Draw a sphere in the Top view.** This is the sun! Notice how the sphere's poles are oriented along the Y axis, top and bottom. If the sphere had been drawn in the Front view, its poles would have been oriented on the Z axis, front and back. In this project it really doesn't matter that much, but in your own projects this kind of attention to detail will be rewarded.
3. **Position the sun at 0,0,0 and make it ten inches in diameter.** As determined at the beginning of this project, this sphere representing the sun needs to be this size.
4. **Temporarily hide the sun.** Do this by selecting the sun sphere, then choose the Hide Selected command in the Selection sub-menu under the Edit menu.



The reason for this step is to allow easier access to the inserted earth / moon shape instance. It will be inserted into the same location in the model space where it is inside the sun. If the sun is present at the same time, it will be more difficult to set up the earth's offset accurately.

4. **Insert the "Earth w/moon rotating" shape.** Switch to the Front view first. This is to maintain the appropriate orientation of the inserted shape. If it was inserted into the Top view, the entire instance would be rotated 90° from that of the shape workspace.



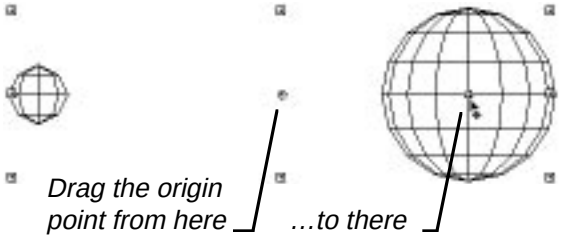
Use the Shapes palette again to do this. The instance will insert into the model space at actual size. Notice that the two objects — moon and earth — are inserted as a single, grouped instance. You cannot access the individual objects within the group at this level, unless you ungroup the instance and break its link with the shape. Notice also that there is one origin point for this instance, and that it is in the geometric center of the group. This is an important piece of information.

5. **Center the instance's origin point on the earth.** As before when you moved the moon's origin point to the center of the earth, this is done using the Object Move tool with the Control key held down.

The object origin is the point on an object that follows its animation "path." If that point is moved, the position of the object shifts relative to its path. This can be useful, but the point cannot move without a conscious decision to do so. Thus, the Control modifier key must be held down to allow you to move the object origin point on objects.

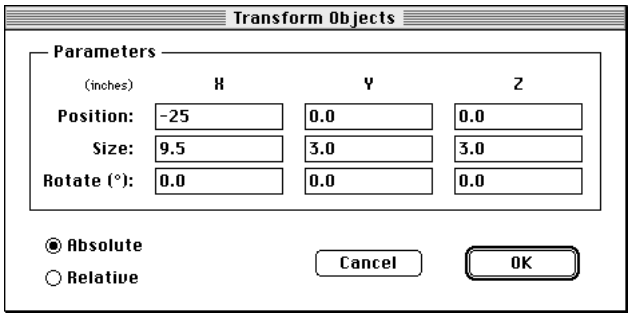
Moving the object origin point

Hold down the Control key while dragging the origin point. Point moves while keeping the instance stationary



Without the Control key, the entire instance would move as you drag its origin point. Remember, you can also hold the Shift key down once you begin dragging the origin point to constrain its movement.

6. **Move the "Earth, w/moon rotating" instance outward to its orbital distance.** Again, the most practical way to do this is through the Transform command. With the mode set to absolute in the Transform dialog, set the instance's position to -25 on the X axis, as shown in the following illustration. This was the earth's orbital radius decided at the beginning of the project.

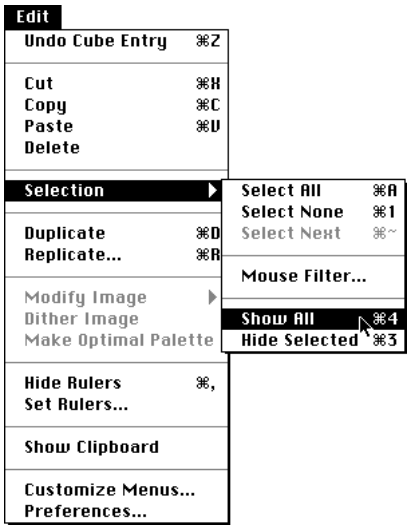


The position of a selection is determined by its object origin point. By moving the object origin point to the center of the earth before re-positioning the earth / moon instance, the earth is

positioned at 25 inches away from the center of the model. If the origin point was at the center of the instance's geometry instead, the earth would not have been positioned accurately.

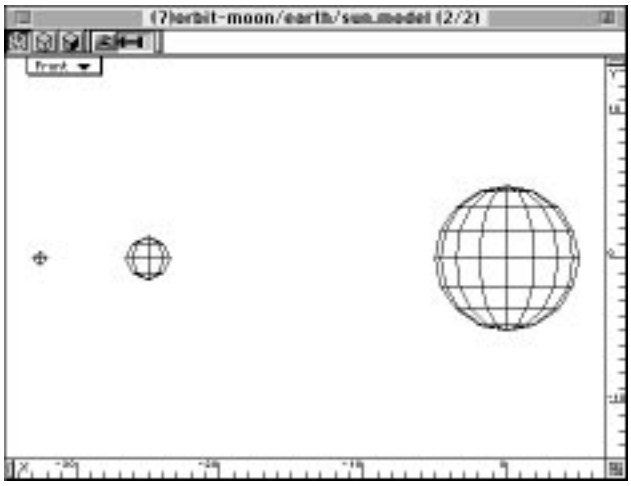
Clicking the OK button in the Transform dialog moves the earth / moon instance out of the active view. It seems to disappear. The instance is still in the model, just not visible at the view plane's current position in 3-D space and in the view's current scale. That will be corrected shortly.

7. **Show the sun again.** Do this by choosing the Show All command in the Selection sub-menu. Alternately, you can use the keyboard equivalent, Command-4 (⌘ 4).



The view redraws to show the sun sphere, right where it was at the time it is was hidden.

8. **Change the view position and scale to see the earth/moon again, along with the sun.** This can be done manually using the View Move and Magnify tools. However, the most practical way is to choose the "Fit Views to All" command in the Windows menu.



The view redraws to show all the objects at a reduced scale in the view. Of course, from your reading in Chapter 3, you understand the mechanics of what has happened with the view, and that the objects have not changed actual size, just that the view has changed the scale at which it is displaying the objects. Notice the rulers also correspond to the reduced viewing scale, but still accurately indicate position of the various objects as you have specified.

Notice also that the sun is shifted to the right edge of the view, which will make it a little difficult to rotate the earth completely around the sun during animation scripting. You could use the Magnify tool to zoom out, as before. But there is an alternate technique which will easily deal with the situation when the time comes.

9. **Switch to the Top view again.** This is in preparation for scripting the earth's orbit around the sun. As you have seen before, in this project the Top view gives the best vantage point for scripting the various objects' orbits.
10. **Move the earth/moon origin point to the sun.** Again, you can simply hold down the Control key and drag the origin point in the Top view to the center of the sun. Remember, the Shift key

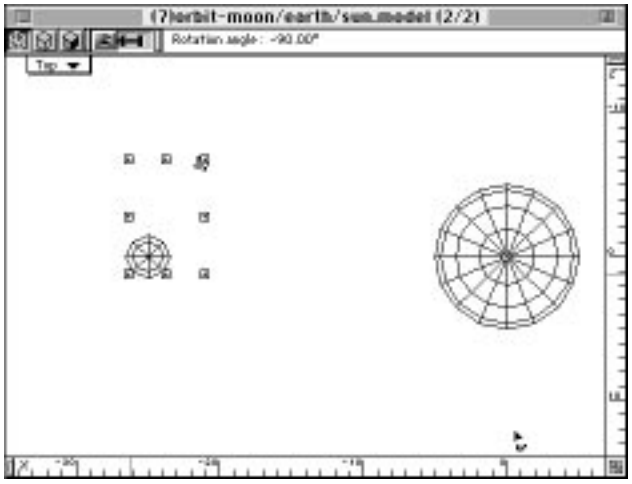
constrains the move so you can be sure it is moved horizontally directly to the center of the sun's sphere.

Now you are ready to begin laying down event markers for the earth's orbit.

11. **Advance the time pointer to 2.25 second mark on the time line.** Of course, you could script event markers at three seconds, six seconds, and nine seconds, just as before. Whereas in previous scripting you have used 120° rotations between event markers, this will be slightly different by scripting four 90° rotations. There is no difference in the behavior, this is just a different example to accomplish the same task.

    Thus, the earth's orbit has a duration of nine seconds; one-quarter of that is 2.25 seconds. Notice that when the time pointer is moved to that point, the moon moves to the position for which it has been scripted in the shape workspace.

12. **Rotate the earth/moon instance 90° counter-clockwise.** With the earth / moon instance selected and the Object Rotate tool selected, click and drag the bottom corner handle downward. The instance's marquee will move out of the view as you drag. Your cursor can even move outside of the window frame, but as long as the mouse button is held down you are still controlling the rotation of the instance in the model space.



Watch the numerical feedback at the top of the window. When it reaches 90°, release the mouse button. The view redraws WITHOUT the earth / moon instance visible and an event marker appears on the Animation palette at 2.5 seconds.

13. **Fit the view to the objects again.** Use the Fit Views to All command again to re-position the view set and show all the objects. This is in preparation for the next rotation.
14. **Advance the time pointer to 4.5 seconds and rotate the instance another 90°.** You now have a cumulative rotation of 180°. Use the Fit Views to All commands again. (By now, view navigation should be routine operations.)
15. **Advance the time pointer to 6.75 seconds and rotate another 90°.** You are almost done.
16. **Option-drag the first event marker at time zero to the nine second position.** The Animation palette will auto-scroll as you drag the event marker. Of course, this is the same technique used on all the previous 360° rotations you've created.
17. **Zoom out to see the entire "Solar System."** Use the Magnify tool with the Option key held down to reduce the view's scale so you can see the whole orbit.
18. **Click on the Preview button.** Ensure that the earth orbits the sun in a smooth motion from beginning to end. Notice how the bounding box marquee for the earth / moon instance grows and shrinks, then shifts sideways. That's the moon orbiting the earth as it follows the earth around the sun. It may be hard to tell if the earth is moving smoothly around the sun with the bounding box continuously changing proportions, but if you've followed the steps correctly, everything should be right.

This ends the modeling phases of this project. Be sure to save the model again before continuing.

Rendering the animation

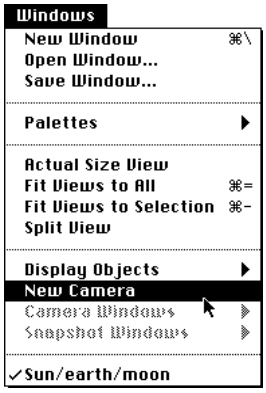
Obviously, the payoff for all your work in this project is to see it in a rendered animation…in motion. No textures have been applied to these objects, so it may be a little dull. But you can always add that later. Note that if no textures are currently applied to an object anywhere on the time line, when a texture is applied at time zero it propagates through all the event markers uniformly. Once a texture is applied, however, changing the texture sets up a texture

metamorphosis action between event markers. THIS IS AN IMPORTANT POINT TO REMEMBER.

The trick for an effective animation is picking the best viewing position. In this project, the camera will not move. Unlike the real world (or should it be real universe), you can go hang stationary in space and watch the world(s) go by. And Einstein said it couldn't be done! Afterward, you may want to script some motion to the camera — a flyby of the moon perhaps? You may get some idea how hard it was to calculate and execute the flight paths that took men to the moon and back in 1969. They traveled a lot farther than in your scaled-down model…and they didn't have computers even half as powerful as yours!

To set up the viewing position:

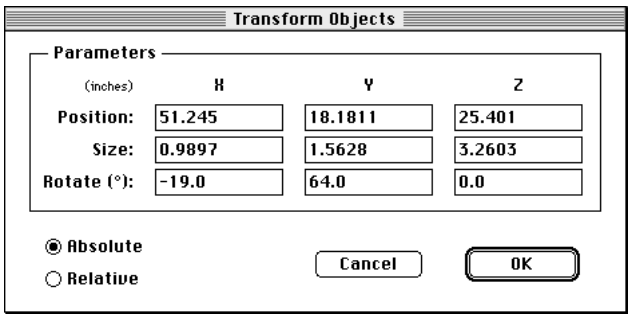
1. **Set the time pointer at time zero.** This is always good practice. It's important to always be aware of the time pointer's position on the time line before you start inserting or moving objects in a model.
2. **Select the New Camera command in the Windows menu.** The most efficient method for setting up a viewing position is to use a camera object.



This inserts a camera object in the center of the active view, and just in front of wherever that view plane happens to be in 3-D space.

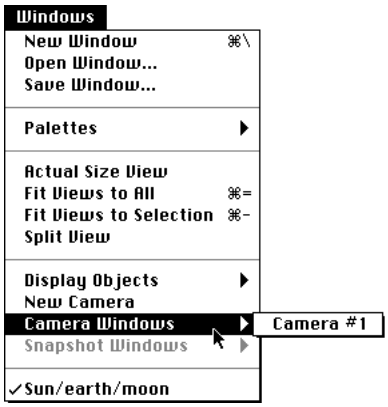
3. **Choose the Transform command.** Let's make this part real easy. Just enter the numbers shown in the following illustration to set

the position and orientation of the camera in 3-D space. This position was arrived at after much soul searching and experimentation — using the "Yeah, that looks about right" approach.



As seen from the Front view, these numbers position the camera object to the left and slightly in front of the sun, but quite a ways above the orbital plane. Seriously, it does give the best perspective view of the earth's full orbit.

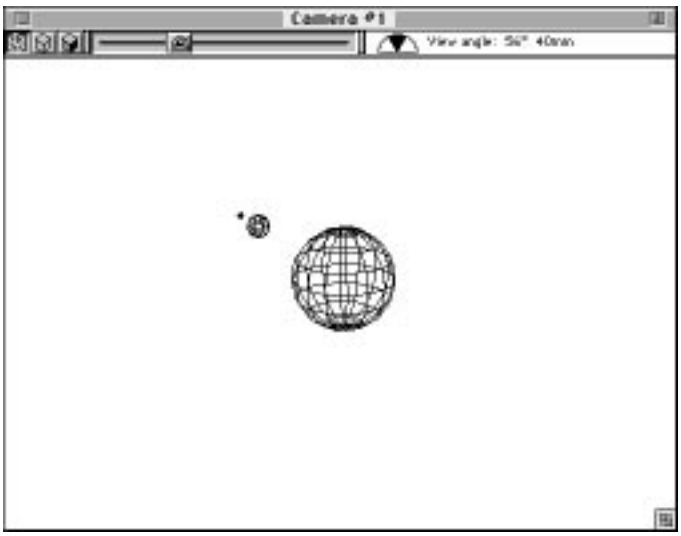
4. **Open the camera object's window.** This can be done by double-clicking on the camera object or by choosing it by name from the Camera Windows sub-menu.



Either way opens a camera window showing the model as if looking through the viewfinder of the camera object. The name of the camera object defaults to Camera #1, but can be changed in the Edit Primitive command's dialog box.

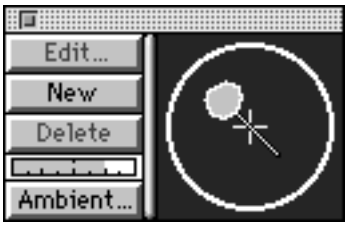
5. **Re-size the camera window and adjust its focal length.** The camera window defaults to a fixed size, which is not wide

enough for this project. Set the size to 500 X 350. Set the focal length to 56° / 40mm.



With the time pointer at time zero, the above illustration shows what you should see in the camera window. The sun is closest to the camera, with the earth and moon behind it. As the earth orbits the sun, it will come toward the camera passing slightly below its viewing position, then around the back side of the sun to its starting position...all in nine seconds.

6. **Set the lighting position.** With the camera window the active view, note that the directional / global light source on the Lighting palette is positioned on the back hemisphere. Hold down the Control key as you drag the light into a new position on the front hemisphere. It should be positioned as shown in the following illustration. Of course, in wireframe renderings, the lighting doesn't matter, but it would be important at higher levels of quality.

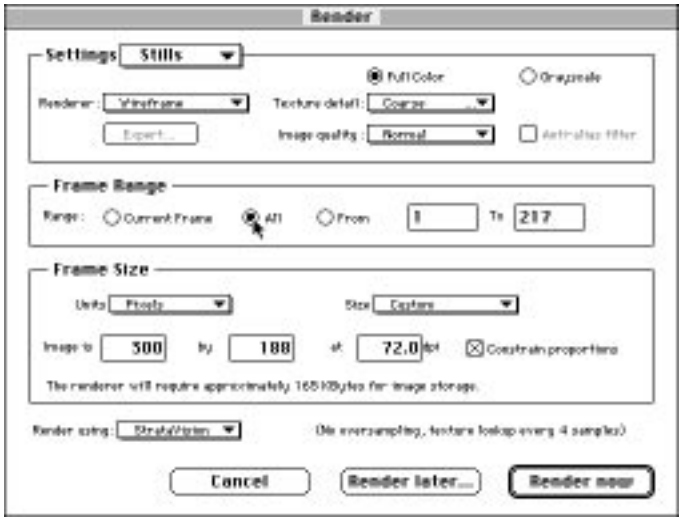


The best arrangement would be to place a point light source in the center of the sphere, so that your sun would appear to be illuminating the earth and moon. You would then need to apply a texture to the sun that turned off its shadow casting (via the checkbox in the Expert level of the texture editing dialogs, remember?) so that light source's illumination could get through the surface of the sphere. This is explained in Chapter 6. You may want to do that as part of the enhancement to this project, along with adding textures. You would also want to delete the default directional light source or at least set it to a low intensity, so it didn't overpower the sun as the primary illumination source for this model.

7. **Render the animation.** It would be best to start with a wireframe version first. Wireframe renders quickly (remember, at 24 frames per second there are 216 frames to render) and it will give you a good sense of the motion (which is the purpose of this exercise). If you are happy with the wireframe animation, move up to the Gouraud rendering method, which gives the fastest smooth shading renderings. If you further apply surface textures with maps representing the earth's and moon's geographies, you will need to use the Phong renderer, which is the real workhorse in animation projects.

   To render the animation, select the Camera tool on the Tool palette. That's the big button at the bottom of the palette. Hold down the Shift key while clicking once in the camera window.

   Clicking the camera tool takes a "snapshot." More technically, it initiates a rendering. Holding down the Shift key while clicking opens the Render dialog, so you can set the rendering parameters before actually beginning the rendering process. Without the Shift key, the rendering begins immediately but will only render a single image at the current time. Note that shift-clicking with the camera tool is exactly the same as choosing the Render command in the Rendering menu. (For other projects, you may also click and drag a cropping marquee while holding down the Shift key.)

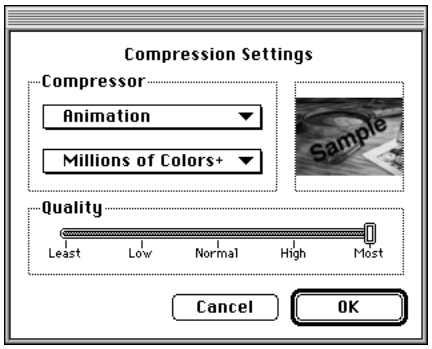


In the Settings section of the Render dialog, set the Renderer field to “Wireframe” and the Image quality to “Normal.” In the Frame Range section, click on the “All” radio button. In the Frame Size section, set the first field of the Image size to “300.” With the “Constrain proportions” checkbox enabled, the second dimension field automatically adjusts to maintain the aspect ratio of the image frame. If the camera window was set to 500 by 350, the second value in the render dialog should be 210. Reducing the frame size increases the rendering speed, takes less disk space, and allows the animation to play faster when complete.

Click on the “Render Later” button at the bottom of the dialog. You may render immediately, if you wish. However, using the Render Later option lets you save a suspended rendering that you may open and begin rendering at your convenience.

Suspended renderings are self-contained, so they do not require the original model to be open during rendering.

The Save As dialog appears for you to designate the name and location where you wish to save the suspended rendering file. This dialog also allows you to choose the file format for saving the animation. The two choices are PICS and QuickTime Movie. Movie format is available ONLY if the QuickTime 1.6 or later extension is installed in your machine's System Folder. (See the discussion on QuickTime support later in this chapter.)



Choose the Movie format and click on the OK button. A second dialog appears allowing you to specify the movie's compression parameters. For this project, just click on the OK button to accept the recommended default settings.

The suspended rendering file is saved. To actually render the animation, just double-click on the file icon on the desktop, or use the Open command in the File menu. The various options and details of rendering are discussed in *Chapter 8: Rendering.*

When the rendering of your animation is complete, you can view it in Vision 3d with an animation window. Animation windows have playback controls that work much like a home VCR. Please refer to

Chapter 1: General Operation for a discussion of the different types of windows used in Vision 3d — the Animation window is explained there in detail.

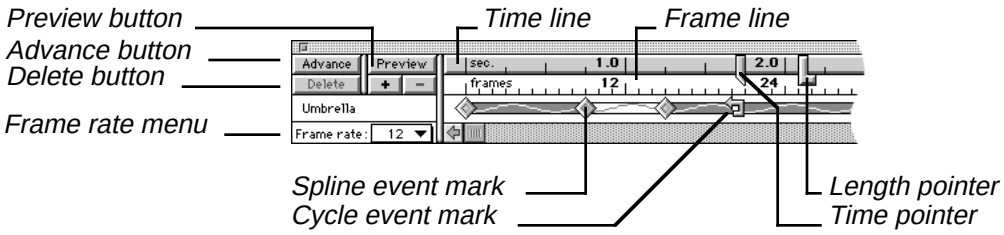
This example has taken you step-by-step through the process of efficiently creating complex motion — scripting simple motions at each level and building up the actions to accomplish the desired motion. The advantage should be obvious: If you want to adjust part of an object's motion (speed up or slow down the orbit of the moon, make its orbit oval instead of round, etc.), you can just go to the shape level where that specific action is scripted to make the change.

This example that you just worked through was itself quite simple, involving scripting the motion of just three spheres. But the concepts presented here are what's really important. Now you must extrapolate and adapt them to your own projects.

The Animation palette

The Animation palette places all controls needed to create an animation in one convenient location. The palette may be switched between scene-based animation and object-based animation, i.e. expanded to show additional data for selected objects.

The Animation palette is designed to float in front of all document windows, but may be covered by other palettes. If another palette overlaps it, the Animation palette may be brought forward by clicking on it.



The Animation palette contains the standard window controls found on all desktop windows, including a move bar, close box, zoom box, re-size box, and scroll bars. These controls should be familiar to you.

The animation palette may be sized and positioned on the desktop as desired. Its re-size box works horizontally only. The close

box hides the palette. It is the same as choosing the Hide Animation Palette command from the Palettes sub-menu in the Windows menu.

The specific Animation controls contained on this palette are as follows:

- **Advance button.** Use to increment the time pointer to the next whole second. If the length pointer is next to the time pointer, it is automatically advanced, too.
- **Preview button.** Use to preview the scripted animation in the model or camera window. This preview is not a rendering, but gives motion to the objects' bounding boxes. It is intended to be only the coarsest kind of preview to help detect gross errors in the motion.

  In complex models, the speed at which the preview progresses can be improved by selecting only those objects that you wish to see in motion. Only the selected objects' bounding boxes become animated. If there is no selection, then all objects' bounding boxes are put in motion, which may slow the running speed.
- **Delete button.** Use to remove an event marker from the Animation palette, and from the affected object's database. This operation is undo-able.
- **Frame rate pull-down menu.** Use to set the frames per second for the document. A new document defaults to 12 frames per second. Thirty frames per second is the rate at which video tape is recorded.
- **Time Pointer.** Use to set the current time. This is done by dragging the pointer with the cursor to the desired position on the time line. As the pointer is dragged, the objects in the model with animation paths will move to reflect their action at that moment in time. Releasing the mouse button causes the model to redraw as it is defined by the animation paths scripted for objects in the model.
- **Length Pointer.** Use to set the length of the animation. This is done by dragging the length pointer with the cursor to the desired position on the time line. The time pointer cannot be moved past this position on the time line, although event markers can. Further, it is the length displayed in the Render dialog. No animation frames can be rendered after this point on the time line, even if event markers are present.

- **Zoom in/Zoom out Buttons.** Use to expand or contract the display of the time line horizontally on the palette. Each click on the plus (+) or minus (-) buttons zooms in or out 2X. The maximum expansion is six times normal, or three clicks on the plus button. The zoom out (minus button) will not contract further than the default normal scale. This feature is useful in situations where you need to position the time pointer and/or event markers more accurately than is possible in normal scale.
- **Event markers.** Use to script the animation paths of object or light sources in a model. Event markers are created as object are positioned, rotated, scaled, or given textures. Event markers are created for light sources as they are positioned, rotated, given gels or textures, and adjusted for illumination intensity.

  Cycle-enabled event markers appear when cycling is enabled for a specified segment of an animation path. This is done through the Cycle command in the Animation palette. The event marker behaves as it would without cycling, but the icon for that marker at which cycling occurs is changed to a folded arrow pointing back toward the left.

Selecting and moving event markers on the palette

Event markers are selected by clicking on them. The inset of a selected event marker changes to a red color to indicate its selection status (black on a monochrome monitor). Multiple event markers may be selected by holding down the Shift key while clicking of each marker icon, or by dragging a marquee around the event markers you wish to select. Multiple selections do not need to be contiguous.

Selecting an event marker DOES NOT automatically move the time pointer to that position on the time line. If you hold down the Command key while clicking on an event marker, the time pointer will move to that position, causing Vision 3d to redraw all open views to show the state of the model at that point in time.

Event markers may be moved along the time line to alter the object's pacing and sequence in the animation. Event markers cannot be dragged from one object line to another.

An event marker is deleted by selecting it, then clicking on the Delete button on the palette. Alternately, you can use the Delete command in the Edit menu or push the Delete key on the keyboard.

Modifier keys that apply to event markers:

A. Shift key — Extends the selection of event markers on the palette. Multiple event markers may be selected at one time for copying, deleting or moving in unison.

B. Option key — Leaves original event marker in the original location and creates a copy of that event marker as you drag it to a new location.

C. Command key — Moves the time pointer to the position of an event marker when you click to select it.

D. Control key — Spaces out or compresses the distance between selected event markers as you drag one of the selected marker. This is useful when you want to open some room for one or more additional event markers, or close the gap after removing some markers.

Quick tutorials

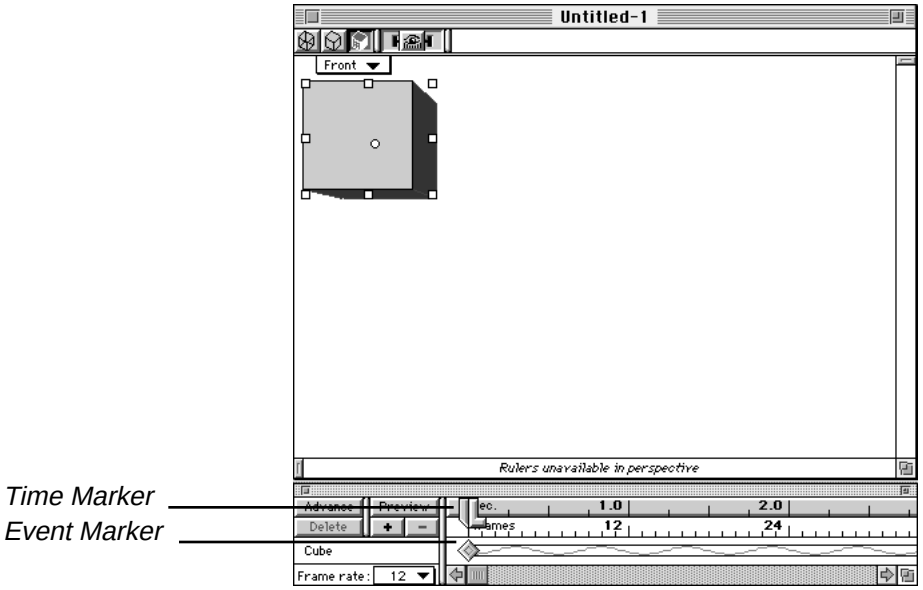
The purpose of the tutorial exercises in this chapter is to familiarize you with the concepts of the event-based animation mechanism in Vision 3d.

Exercise 1: Simple motion

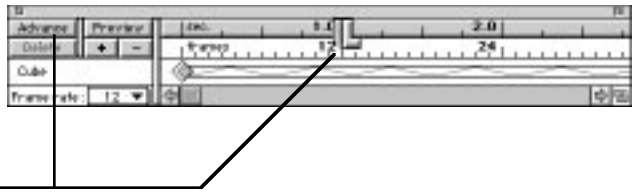
In Exercise 1, you will learn how to lay down event markers on the Animation palette to animate a simple object. This is fundamental to all animation scripting operations in Vision 3d.

1. **Open a new model.** Set the display method to Shaded Surfaces and the three-position lens control to Normal. Make sure you're in the Front view.
2. **Draw a cube in the upper left corner of the model window.** The size is not important, just make is a comfortable size for the view, as shown in the illustrations on the following page.
3. **Display the Animation palette, if it's not already visible on the desktop.** Pull down the Windows menu and choose Show Animation Palette from the Palettes sub-menu. Vision 3d shows the floating Animation palette where is was last positioned, usually at the bottom of the screen. Notice how each object, when selected, appears on the timeline extending to the right the

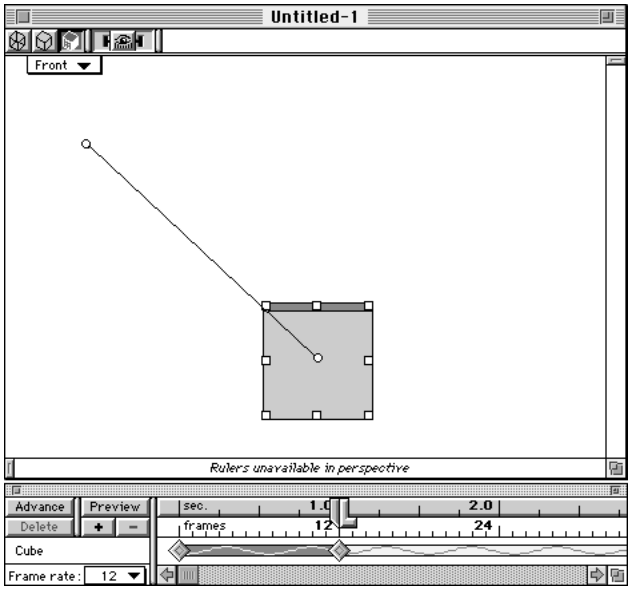
object's name. Notice also that the animation time indicator is currently at time zero. The single event marker on the timeline indicates that the selected object in the model exists in its current position from time zero on without any changes.



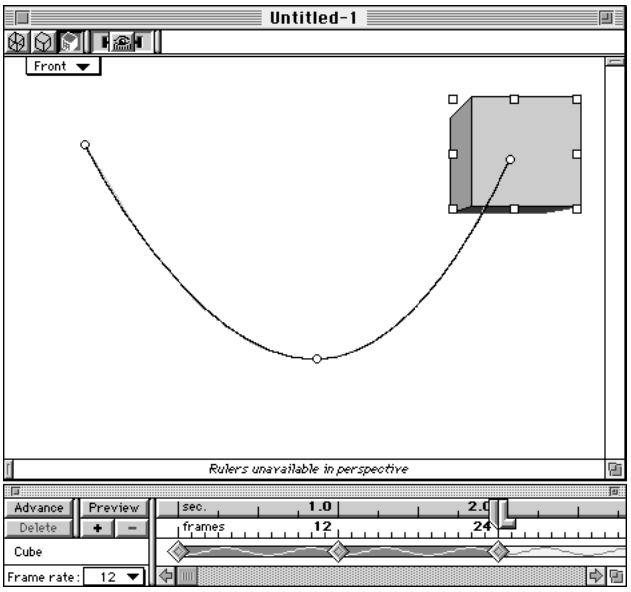
4. **Click on the Advance button.** Vision 3d responds by moving the time indicator to the one-second position on the time line.



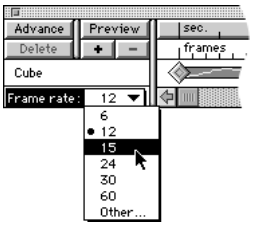
5. **Drag the cube to the lower center of the Front view.** Be sure to choose the Pointer tool to do this, if it's not already chosen. Notice that when the cube is repositioned, Vision 3d creates a new event marker at the one-second time for the cube. The new marker indicates that at one second, the cube is in the position you have specified.



6. **Click the Advance button again.** This moves the time pointer to two seconds.
7. **Drag the cube to the upper right corner of the Front view.** Again, Vision 3d creates a new event marker at the two second position for the cube. This event marker will be the last one in your animation.



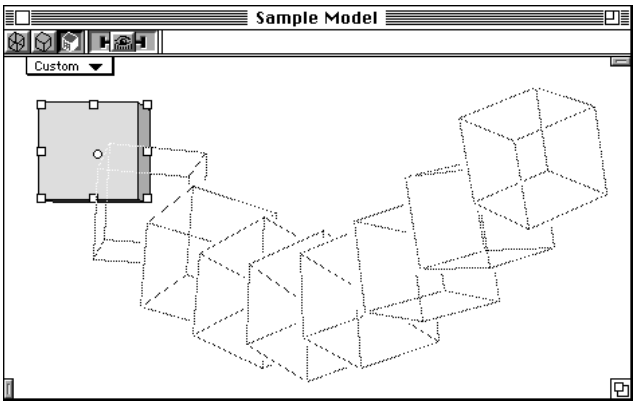
8. **Set the Frame Rate field to 15.** This indicates to Vision 3d that you want to render 15 frames for each second of the animation. Using as time and position references the three event markers you have scripted, Vision 3d automatically calculates the in-between positions for the cube, i.e. — those frames that come between the event markers.



The resulting animation will contain a total of 30 frames. Note that the starting frame at time zero is numbered frame #1 and the last frame is numbered frame #31.

9. **Click on the Preview button to see your animation in the model window.** Vision 3d plays a simple “bounding box” preview of the animation.

As the time pointer is dragged or when the Preview button is clicked, animations play as bounding boxes in the modeling windows



From this very simple exercise, you’ve learned how to set event markers, edit the frame rate and preview an animation. The next step in the process is to actually render and save each frame of the animation sequence to your hard-disk drive so they can be replayed for viewing.

You’ll tackle that process of rendering an animation in the next exercise. But before you get to that point, the next exercise will have you try some more advanced animation techniques.

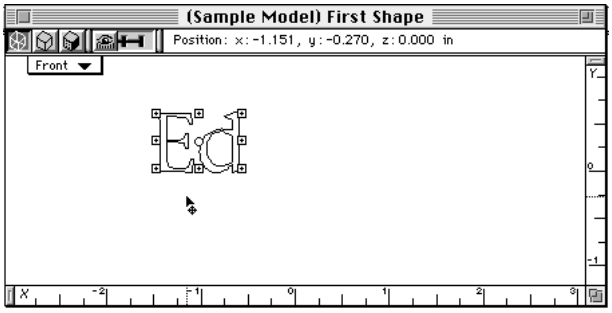
Exercise 2: A flying logo

In this exercise, you’ll learn to do a more complex animation. The purpose is to show you Vision’s hierarchical animation structures, with camera objects, and with the concept of rendering a draft version of the animation to check for problems.

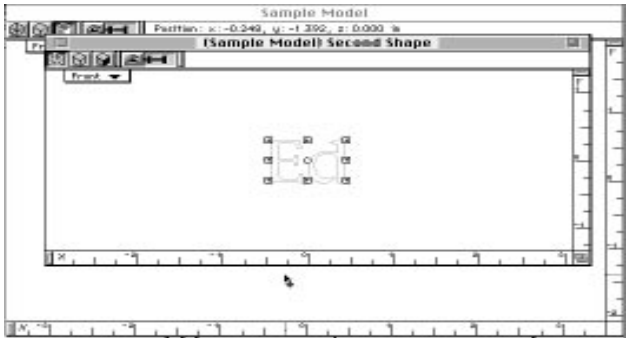
1. **Open a new model.** Set the three-position lens control on Orthographic and the view position at Front, if not already using those setting. Just in case, also choose the Actual Size command in the Windows menu. This resets the window scale to 1:1 and moves the view set center to 0,0,0 in the model space. It’s a good place to begin.

Choose the display method you are most comfortable with. Wireframe is the fastest for redrawing the model views, but shaded provides the most information.

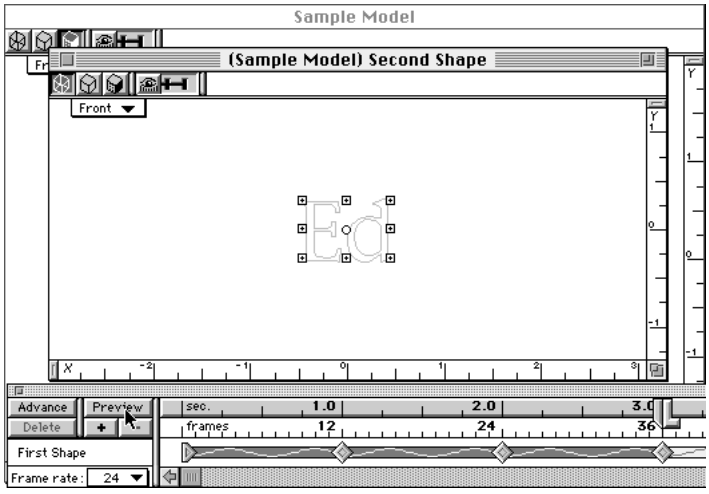
2. **Create a new shape.** Name it “First Shape.”
3. **In the First Shape workspace window, create a 3-D object of your given name.** Use the text tool to do this. In the text dialog, choose a font and style, then type in your first name. Scale the 2-D text objects so they’re about an inch high. Using the Extrude modeler, extrude to a depth of one-half inch.



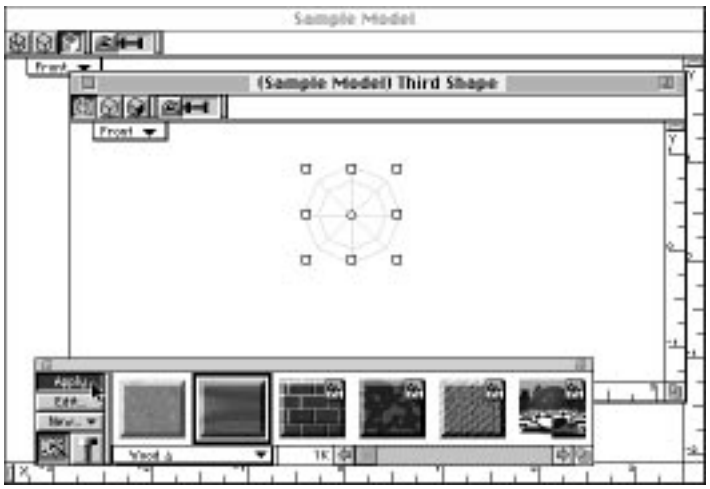
4. **Apply the Blue Enamel texture to your 3-D name.** Still in the shape workspace, choose the texture and apply it to your 3-D name. If you want to see how the texture looks on the object, do a quick preview rendering now.
5. **Close the shape workspace window.** Click on the close box in the upper left corner of the workspace window. Vision 3d returns you to the model window, which at this point in the exercise is still empty.
6. **Create another new shape.** Title this one “Second Shape.”
7. **Insert the First Shape into the Second Shape.** While in the Second Shape workspace, choose First Shape on the Shapes palette, then click on the Insert button. Vision 3d inserts an instance of your 3-D name (with texture) into the Second Shape’s workspace.



8. **Still in the Second Shape workspace, script a three-second animation of your name rotating 360 degrees counter-clockwise.** Display the Animation palette, if it's not already showing on the desktop. Click on the Advance button to move the time indicator to the one-second point. Select the Rotate tool from the tool palette. With your name selected, click on any corner selection handle and rotate your name counter-clockwise a minus 120 degrees. Then click on the Advance button again, and rotate your name a further -120 degrees counter-clockwise. Finally, click once more on the Advance button, and rotate your name another -120° counter-clockwise. You should see three event markers on the First Shape timeline; one each at the one-, two- and three-second points. Check your animation scripting by clicking on the Preview button. The bounding box preview should make a 360-degree rotation.



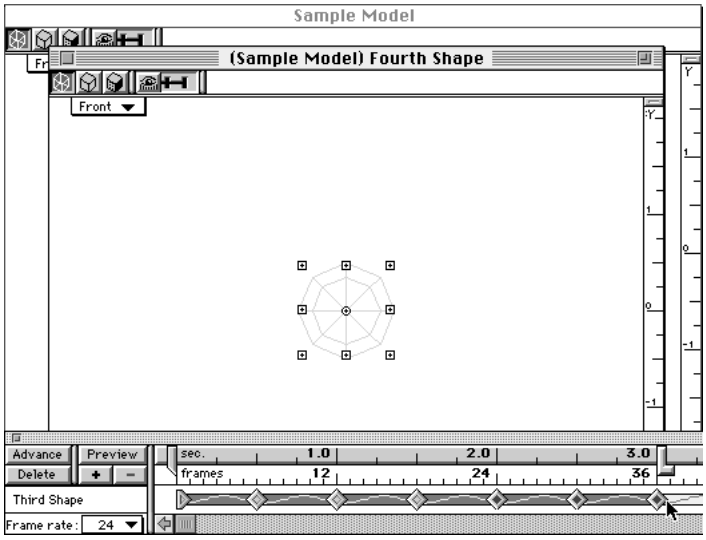
9. **Close the Second Shape workspace.** This would be a good time to save the model. Use any name you like, such as "Flying Logo - my name."
10. **Create a new Shape.** Title this one "Third Shape."
11. **Draw a Sphere in the Third Shape workspace.** Its size should be about two inches in diameter.
12. **Apply the procedural Wood texture to the sphere.** If you want to check this before moving on, do a quick preview rendering now.



13. **Close the Third Shape workspace.** Save the model again.
14. **Create a new Shape.** Call this one…you guessed it, "Fourth Shape."
15. **Insert the Third Shape into the Fourth Shape.** Do this just as you previously inserted the First Shape into the Second Shape, using the Insert button on the Shapes palette. Refer to step 7 if necessary.
16. **In the Fourth Shape workspace, script a 3-second animation of the sphere rotating a total 720 degrees clockwise.** This is very similar to what you did in step 8, except the object is a sphere and you will script it to make two full rotations in three seconds. Refer to step 8 if needed.

Because the sphere will rotate twice as fast as the object in step 8, you must lay down an event marker every one-half second instead of every second. To do this, simply click on the current time indicator and drag it to one-half second. Then rotate the sphere 120 degrees clockwise. Then click-and-drag the current time indicator to one second. Repeat the rotation. Then do the same with the current time pointer at 1.5 seconds.

Now, instead of creating three more event markers, copy the three you have already made, and use them again at different time. To do this, click and drag a marquee around the three event markers on the Third Shape timeline, so all three are selected. You know they are selected because each has a red center (for grayscale monitors, the center is dark gray; on monochrome, the center is black).

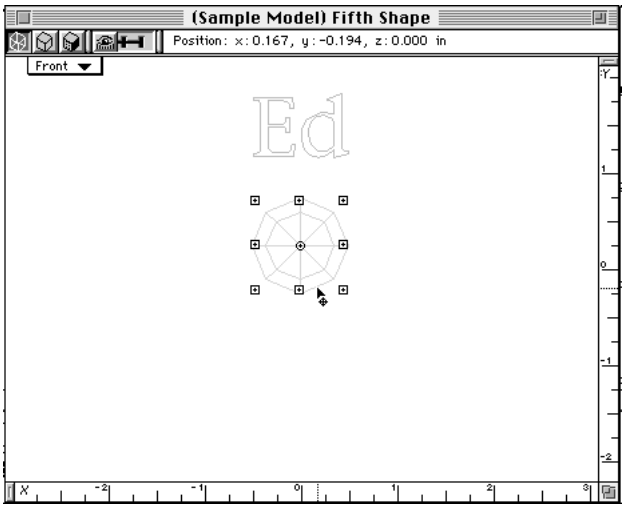


Then, holding down the Option key, click-and-drag the three event markers to duplicate them on the timeline. Drag them to the right until the event marker on the far right side is aligned with the animation length indicator at three seconds. Then release the mouse button. The result gives you six event markers

on the Third Shape's timeline, with marker #4 a duplicate of marker #1, #5 a duplicate of #2, and #3 a duplicate of #6.

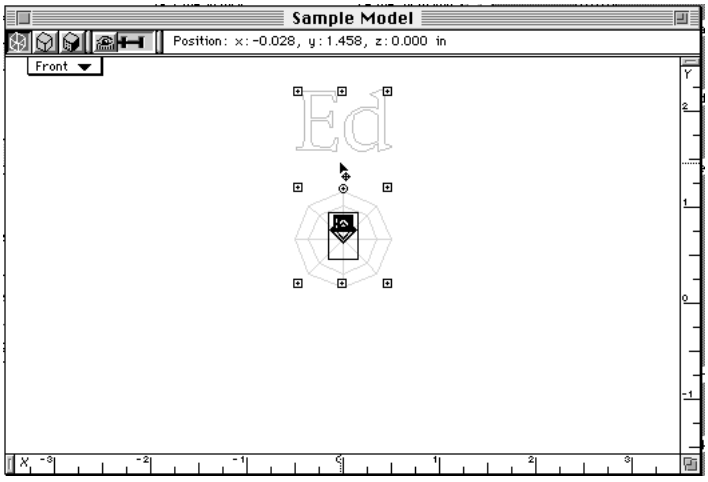
To check your work, preview the animation in the shape window. The bounding box should rotate two full times over the course of the three-second animation.

17. **Close the Fourth Shape workspace.** Time to save the model again.
18. **Create a fifth new Shape.** I think you know what to call it.
19. **Insert the Second Shape and the Fourth Shape into the Fifth Shape.** Refer to step 7 as needed.
20. **Assemble the instances together.** Position the second and fourth shapes as shown in the following illustration.



21. **Close the Fifth Shape workspace.** Save the model again.
22. **Insert a new camera into the model window.** Choose New Camera from the Windows menu.

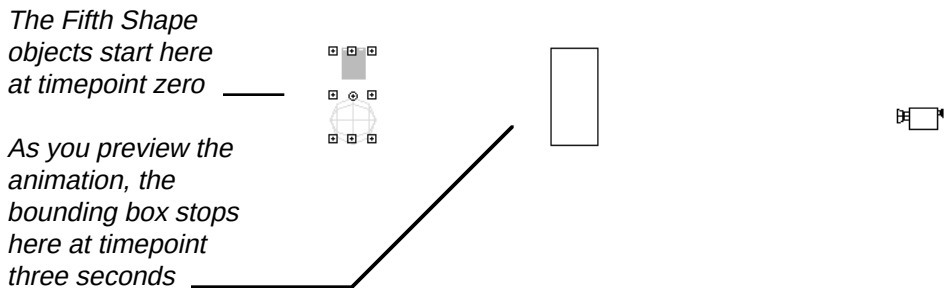
23. **Insert the fifth shape into the model window.** By now, this should be "old hat" for you. Do it just like you've done with other shapes.



24. **In the model window, switch to the left view.** Use the "Fit Views to All" command, if necessary, so you can see the camera object and the fifth shape instance in the Left view. If the camera

is too close to the objects, they may not be fully visible through the camera window. In this case, you may want to move the camera farther away from the objects.

25. **Script an animation that moves the name and sphere objects toward the camera.** Be sure the instance is selected. (Since instances of both objects were inserted into the fifth shape, and the fifth shape in turn was inserted into the model window, clicking on either object will select them as a group.)



By now you know how to script an animation. You probably need only two event markers here — one at time zero for the beginning position, and another at three seconds to set the finishing position. Since the first event marker at time zero was created automatically when you inserted the Fifth Shape into the model window, you only need lay down the finishing marker. Move the time pointer first (to the three-second time), then move the Fifth Shape's instance to its finishing position. That's it! The event marker is created and the animation is ready to fly.

Preview the animation in this view before going on. Also remember to save the model again.

26. **Double click on the camera object.** Vision 3d opens the camera window which displays a view from the camera object's position. You should see something similar to the following illustration

27. **Render a draft of the animation.** It's generally a good idea to render a draft of any animation to check for problems before committing to a final, high-quality rendering.

    Be sure the camera window is the active window. Pull down the Rendering Menu and choose the Render command. In the Settings menu of the Render dialog box, choose "Movies • Draft" from the pull-down menu. This sets up a complete definition of parameters for your rendering. In the Frame Range area, click on the "All" radio button, if not already selected, to render all frames of the animation.

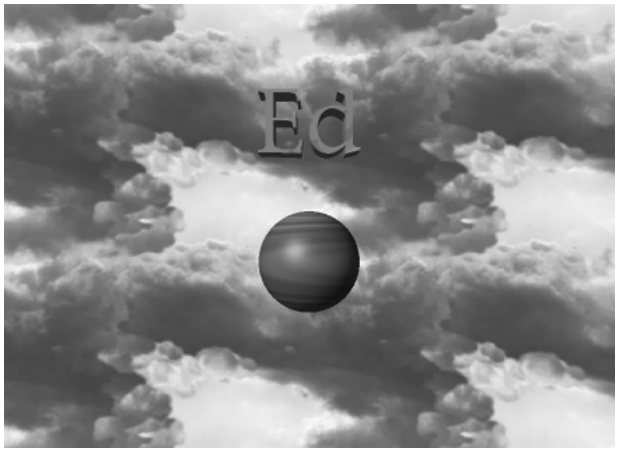
    Then click on the "Render Now" button. Vision 3d opens a Save As dialog, allowing you to name the animation file and choose a file format and location in which to save it. If you have Apple's QuickTime system extension installed, save your animation in QuickTime Movie format.

    Clicking on the OK button presents a second dialog asking for movie compression settings. Go with Animation and 256 colors as the compression settings. If you don't have QuickTime installed, save the animation as a PICS file with 8 bits / pixel resolution, optimal palette and dithering enabled as your options.

With either format, once you click on the OK button, Vision 3d will begin to render each of the frames and automatically save each to the hard disk location you chose.

When the rendering finishes, play back the animation. Check it with a critical eye. Are the objects placed too close together, causing them to intersect as they rotate? Do they move too close to the camera as the animation finishes? If you see these or any other problems, now is the time to correct them. But remember, this animation is hierarchical; specific animation events take place in specific levels. If you edit the animation, be sure to go to the proper workspaces in order to preserve the hierarchy.

If you want, you can render a final high-quality version of this animation. You may want to put a background behind the objects for the final rendering. If so, pick a background environment from the Environment Effects dialog window which is reached through the Rendering menu.



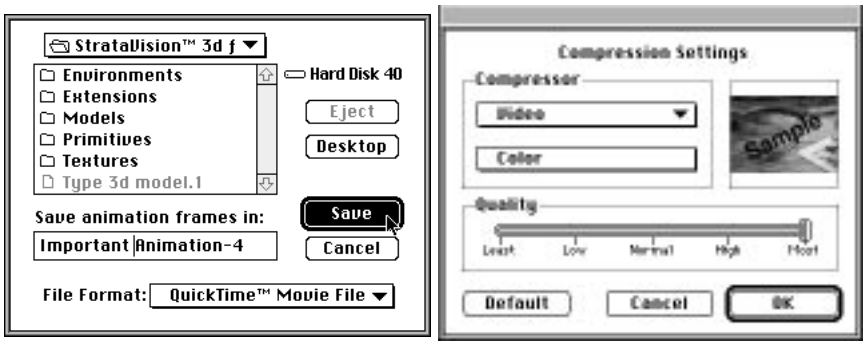
Render your final version using the Movies•Better rendering setting. Before initiating the higher quality rendering, go to the Rendering Effects dialog window and disable all the Object Effects EXCEPT Surface Mapping, Bump Mapping and Smooth Shading. Keep in mind that rendering a high-quality final version of a still shot or an animation always takes longer than rendering a preview or a draft version. But once you've seen a preview or draft and know

that everything is ready, you're ready to commit to a final rendering. And the final results will be worth waiting a little longer.

QuickTime support

When saving an animation in QuickTime format, a second dialog appears in which you are given a choice of the compression algorithm to use and how much the file is compressed. A sample image is displayed in the QuickTime compression dialog that previews the effects the various settings will have on your animation frames. You have the option of using a QuickTime compression method appropriate to the format — either during the creation of the animation or when switching formats with the "Save A Copy As" command.

You should be aware that whenever QuickTime compression is used to save a file, the QuickTime extensions must also be present to open the file again or you will not have access to the compressed file.



The compression settings are a key part of the rendering process when saving in QuickTime Movie format. They can greatly affect the results. And there is also an interplay between the various compressors and the amount of compression. Vision's default settings use the Animation compressor, with Millions of Colors+, and Most quality. This is obviously biased toward quality over file size. However, it is important to have a good understanding of what the compression settings are, and what affect modifying each one of them has for the different compressors. Once this is achieved, you can make more intelligent choices of what compression settings to use, based on the intended use. Of course, you're encouraged to

experiment with different settings, to get a feel for what the limitations and advantages are for each compressor.

An explanation of the different compressors and their recommended use is included later in this section of the chapter. To better illustrate the tradeoffs between compression settings and intended use, here are a few example scenarios:

1. Suppose you want to send a short (approx. ten seconds) preview animation to a client for approval over a 9600 baud modem. To do this, it may be acceptable to greatly reduce the image quality, so that a practical transmission time can be achieved. In this case, low quality settings would be used, even if this incurs noticeable blocking artifacts. This is an example of "time is of the essence."
2. Suppose a colleague wants to add an animation at the last minute to an already-completed presentation for a small meeting. In this case usage of average compression settings would be a reasonable trade-off, as this would yield acceptable image quality without a large file size. Since the animation will be seen by a relatively small number of people only once, it isn't essential that it be high quality.
3. The final example is for CD-ROM production. In this case, you would want to get the best image quality possible. You would use high quality settings without undo regard for file size, since the CD-ROM has such a large capacity and it will be distributed to many people.

After an animation is rendered in the QuickTime Movie format, if it does not play fast enough for your purposes, other QuickTime utility-type applications are available to re-calculate it for improved compression and better performance.

One of the additional compressors available is called frame differencing. When saving in Movie format, the individual frames of an animation being rendered are compressed frame by frame using one of the standard compressor algorithms. Frame differencing is not possible at that time. Once a movie is saved, additional compression methods can be performed, either by using the "Save A Copy As" command in Vision 3d or through the other applications intended for this purpose

Frame differencing stores a complete frame at regular intervals (usually each second), with in-between frames containing only the changes in the image from complete frame to complete frame and using the last whole image as a reference. Each complete frame starts

the process over again. Depending on the number of frames per second, the data savings can be substantial. By storing the difference between frames rather than a full image every frame, the animation file can be much smaller. And playback is usually faster. However, actual compression ratios and improvements in performance are highly dependent on the type of images in the animation.

There are a couple of considerations when using frame-differenced compression. Since the image on the screen must be sequentially constructed from one frame to the next, visual anomalies may occur in the animation playback when frame skipping is allowed. If this is a problem, disable the frame skipping option in Vision's Animation menu to force QuickTime to play every frame of the animation. While it may play slightly slower in this mode, the frame-by-frame images will always appear intact this way.

Using QuickTime for still images

QuickTime compression for PICT and PICS files is enabled by a check box in the secondary Save dialog box, but is only available if the QuickTime extension is present. The compression method for these files uses the Joint Photographic Experts Group (JPEG). This is a "lossy" method of compressing images. Compression is achieved by removing redundant or imperceptible data from the image, which must be reconstructed upon expansion by extrapolating the remaining pixels. Depending on the amount of compression, JPEG can affect an image's appearance. However, Vision 3d gives priority to maintaining image quality and does not compress its files as tightly as may be possible.

No compression is available for still images saved in TIFF format.

Playing a compressed PICS file from hard disk is not recommended without specific hardware support — i.e., a NuBus card containing the JPEG compression / decompression code in ROM. Without hardware support, the time required to expand each frame virtually eliminates any possibility of smooth, flowing visual motion.

On the other hand, compressed PICS animations are significantly smaller and easier to transport…and they work just fine when played from memory. In Vision 3d, this is done by first opening the file, then using the "Playback from Memory" command in the Animation menu.

Explanation of QT compressors

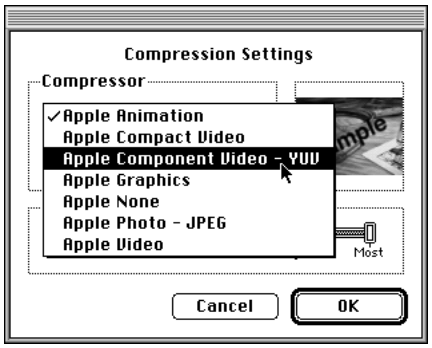
Several software compressors are provided through the QuickTime extension. These are listed and chosen in the pull-down menu of the Compression Settings dialog. Each compressor was designed for a specific type of use. Using a compressor not intended for your needs is not wrong, per se, but using the right compressor can significantly improve playback performance.

The following explanations are provided for background:

- **Video compressor.** Of all the available compressors, the Video compressor is best suited for capture and compression of analog video, high-quality playback from hard disk and moderate to high quality playback direct from CD-ROM. This compressor can play back at frame rates of 10 fps or better, and can be transcoded to achieve further compression and to permit the ability to play back directly from CD-ROM. A file compressed with this method is re-compressible with minimal or no quality degradation, allowing digital effects processing and editing to be applied in repeated passes.
- **Compact Video compressor.** This compressor is an extension of the Video compressor. All characteristics and uses identified previously for the Video compressor also apply to this compressor. However, it achieves a better image quality for a given compression level because it has been optimized for 16-bit color ONLY (thousands of color). The dithering algorithm is significantly improved because of this optimization.
- **Animation compressor.** The Animation compressor is best suited for images that are originally created in digital form, such as rendering. It performs best on images that have no noise. The Animation compressor can be used to compress digitized analog video, but it may not produce high compression ratios from that type of material. If the Animation compressor is used for captured analog video, it will produce very good picture quality which can be played from a hard disk. It is unlikely that this compressor will be fast enough to compress and/or play analog video sequences from CD-ROM at sufficient speed.
- **Photo compressor.** Of all the currently available compressors, the Photo compressor produces the highest quality images at the best compression ratios. It is better know as JPEG compression. However, it is not currently fast enough to play back moving image sequences with sound at acceptable frame

rates. Because it does achieve such a good result at high compression ratios, the Photo compressor can be used effectively as an archival method for video clips with high disk space requirements. JPEG is supported on several platforms, making this compressor a viable interchange format between computers of different manufacturers.

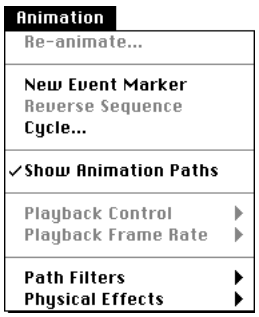
- **None compressor.** The None compressor is a very good format to use for acquisition of analog video. It is also good for digitally-generated animations. It has the advantages of having a very good image quality (since no compression is applied), it can capture in real-time with sound on higher-end machines, and it can be further transcoded off-line to produce movies with much better compression ratios. The disadvantage is the high disk space requirements.
- **Graphics compressor.** The Graphics compressor is a good format for acquisition of analog video when there is a need to have very good picture quality for playback on eight-bit displays. To achieve the best results, it is necessary to capture the video at 24 bits per pixel, apply error diffusion dithering as it is converted to eight bits per pixel, then compress the result. Although the Graphics compressor was intended primarily for use with 8-bit source material, it can be effective for video in this context. It does not achieve high compression ratios for video, so it is not currently suitable for playback directly from CD-ROM. It will work well for play back from hard disks.
- **Component Video —YUV compressor.** By default, the YUV compressor does not appear in the standard compression type list. If you hold down the Option key while clicking on the pull-down menu, it will appear.



The YUV compressor is somewhat specialized. It is useful with certain video-editing solutions. It makes a good intermediate storage format if you are applying multiple effects or transitions to an animation. The compression ratio is always 2:1 and the image quality is extremely high. Frame differencing is not supported.

Animation extensions

Installed animation extensions are accessed by commands that appear in the Path Filters or Physical Effects sub-menus of the Animation menu. Vision 3d includes some extensions in the package as part of the standard product. As other extensions become available and are installed in the Extensions folder, they also will appear as commands on the Animation menu.



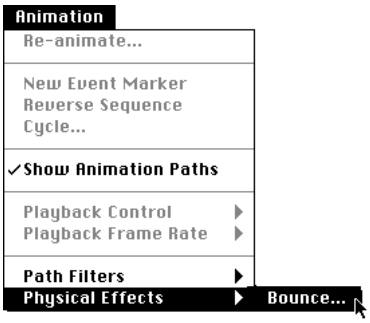
Modeling extensions provide their own dialogs and menu bar allowing the construction of new objects, for converting existing objects to a type supported by the modeler, or for further editing by the same modeler. Animation extensions are similar in behavior.

All necessary tools required to perform the work for which the modeler is designed are also provided in the dialogs. When editing the object is complete, the extension inserts the object back into the model space with the changes you specify.

The standard animation extensions that ship with Vision 3d are discussed in this section of this chapter in alphabetical order. Other after-market or third-party extensions will include their own documentation explaining how to use the animation features provided.

Bounce

When present in the StrataVision 3d Extensions folder, the command to use the Bounce extension appears in the Physical Effects sub-menu of the Animation menu.



If the Bounce command does not appear in the Animation menu, the extension is not correctly installed.

Description

The Bounce extension creates the animation path to bounce an object. Event markers are generated and placed in the appropriate order and position on the selected object's time line causing the requested bouncing motion. Note that there is no collision detection; it is your responsibility to specify the parameters so that the object's bounce appears correct for the scene you have assembled.

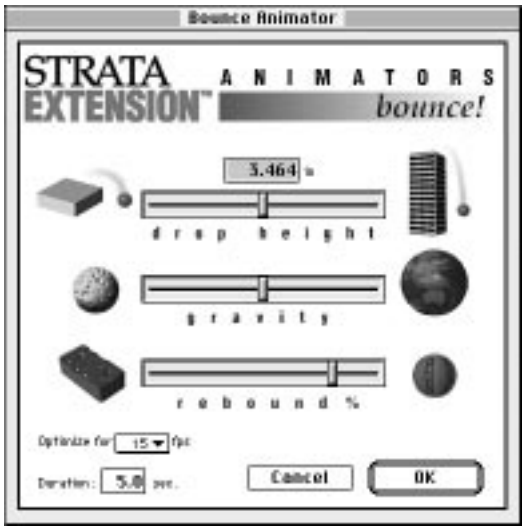
The advantage of the Bounce extension is that it performs all the calculations to automatically create a realistic bounce motion for a selected object. The event markers are generated and laid down on the time line for that object in the Animation palette. The bounce calculations even extend to setting the motion controls for each of the event markers found in the Event Options dialog, i.e. — spline or linear, spline tension, ease in / ease out, etc.

Only one object can be "bounced" at a time. If an animation path is already defined for an object, it will be fully or partially replaced for the specified interval with event markers generated by this extension. If the original animation path extends longer than the one scripted for the bounce effect, those event markers after the bounce interval are left intact. However, the combined animation path may

become quite erratic during transition from the new path created by the Bounce extension back to its original path.

USING BOUNCE

To use the Bounce extension, in the model or a shape workspace, you must already have an object drawn or inserted in the workspace. First, position the time pointer when you want the bounce to begin and select the object. Then, choose the Bounce command from the Animation menu. A Bounce dialog box opens to set the motion parameters.

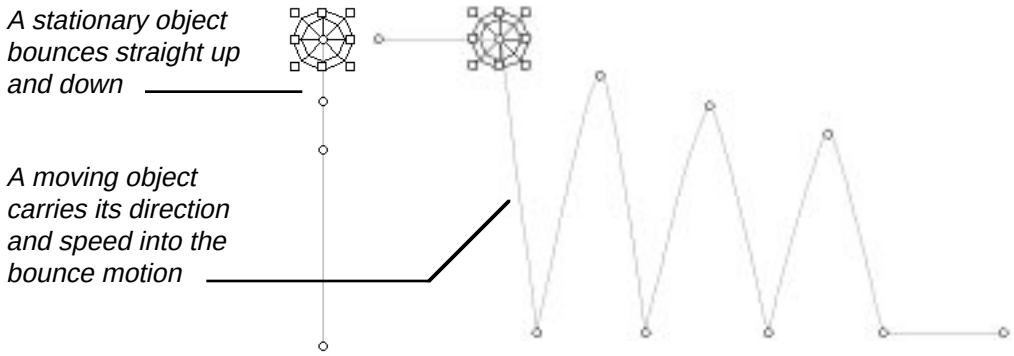


Three slider controls allow you to set the drop height, the gravity strength, and the rebound percentage (a.k.a “bounce-osity”). Fields are provided to set the frames per second and the duration to be used in the bounce calculations. The preferred frames per second setting is important because of temporal aliasing concerns — a mismatch between the object’s bouncing motion and the fineness at which the tweening occurs during rendering can cause jumps in the bounce. The duration determines the maximum time in which the bounce motion will decay to zero. It may stop bouncing sooner than the duration settings, depending on the drop height, gravity, and rebound, but it will never go longer than the duration specified.

A field is also provided to enter the drop height numerically for greater accuracy. Note that the drop height is measured from the object's current position at the time this extension is called.

Carrying object motion into the bounce

If an object is stationary at the time the Bounce extension is called, it will bounce straight up and down. If the object is moving at the time it begins bouncing, that motion (both speed and direction) is added to the bounce calculations. The same motion is maintained after the bounce decays, unless other event markers are already defined to constrain its motion in a different way.

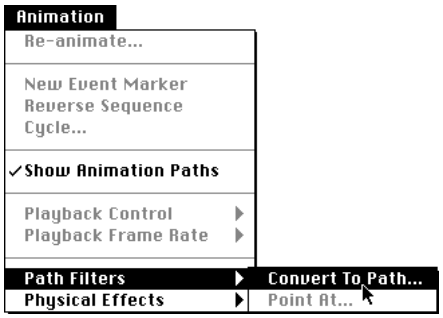

A stationary object bounces straight up and down
A moving object carries its direction and speed into the bounce motion

Of course, once the event markers are in place and the animation path is displayed in the active view, the path is just like any you would create manually. If desired, you can move the markers on the palette or drag the anchor points on the path on any axis to fine-tune the motion any way you want.

For example, suppose you want a ball to bounce down some stairs. There is no provision in the Bounce extension to set a bounce height that changes, like it would on stairs. But after the Bounce extension is done, you can click and drag the individual anchor points on the animation to match the step on a set of stairs, as shown in the previous illustration.

Convert to Path

When the Convert to Path extension is present in the StrataVision 3d Extensions folder, the command to open the Convert to Path extension appears in the Animation menu.



If Convert to Path does not appear in the Path Filters sub-menu of the Animation menu, the extension is incorrectly installed.

DESCRIPTION

The Convert to Path extension converts any 2-D line to an animation path. The line can be drawn in Vision 3d or imported from another source. It can be an open line or a closed polygon, or even a closed bézier spline. Once the line is converted to a path it is no longer an object, and thus does not exist in the model as a physical entity.

The extension allows you to assign a shape, camera object or light source to the new animation path. The assigned object may already exist in the model, or can be initiated from within the extension. Convert to Path also allows you to set a start time when the object will begin its motion on the assigned animation path, and a duration for the motion. These time options are particularly useful when you're coordinating motion scripts within complex animations

which have more than one object moving simultaneously. Only one object can be assigned to each path. But the line used to make any path can be duplicated and re-used multiple times as paths within a model.

Using Convert to Path

To use the Convert to Path extension, in the model or a shape workspace simply draw or import a 2-D line you wish to use for an animation path. You can convert your line to bézier in the 2D Sculpter before using the Convert to Path extension, thus allowing you more flexibility when creating your animation path. Then, with the line selected, pull down the Animation menu and choose the Convert to Path command from the Path Filters sub-menu. Vision 3d opens the extension's dialog box.

Additionally, when the Convert to Path dialog opens, the standard Vision 3d menu bar is replaced by one that supports the extension.

The extension's menu bar contains only three menus. The first two are dimmed because none of their commands are active while in

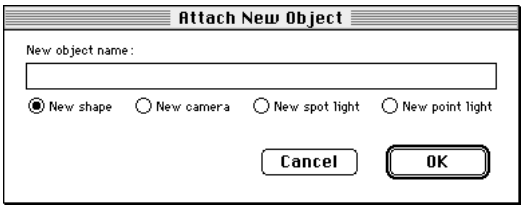
the extension. The Undo command in the Edit menu is the only usable menu command in this extension.

If no objects exist in the model that can be assigned to the path, the dialog opens with an empty directory. If one or more objects that can be assigned to the animation path already exist in the model, it opens with the objects listed in the dialog.

To attach one of the listed objects to the animation path, choose that object on the list and click on the OK button. The chosen object is attached to the path at the beginning point of the path.

The direction of movement along the converted animation path is determined by the direction that it was drawn. In the case of closed polygonal or bézier objects, the object travels from the beginning vertex point to the ending vertex point. Of course, you can change the location of those points on the object with the 2d Sculpter modeling extension before converting it to an animation path.

Regardless of whether objects are listed in the dialog window or not, you have the option to initiate new objects from within the extension. Clicking on the “New Object” button opens the Attach New Object dialog box.



By clicking on the appropriate radio button, you can initiate a new shape, camera object or light source. If you are initiating a new shape or camera object, you must also give it a name. In the case of a new shape, a cube primitive is automatically drawn as a "place holder" in the shape's workspace. You can replace the cube with your own object(s) later.

Clicking on the OK button inserts the new object into the model and attaches it to the animation path at the beginning point of the path.

Starting and Duration times

The value in the Start Time field is the number in seconds on the animation timeline at which the assigned object appears (or is born) and begins moving on its animation path. The value in the Duration field is the length of time in seconds over which the motion on the animation path will occur.

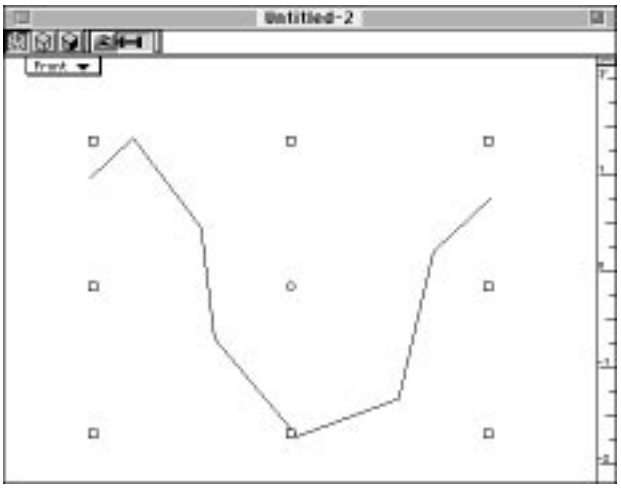
An example

We'll script a short animation using the Convert to Path extension.

1. Open a new modeling window.
2. Create a new shape by clicking on the Shapes icon on the Textures / Shapes palette; then click on New. Call the new shape "Sphere," and click OK.

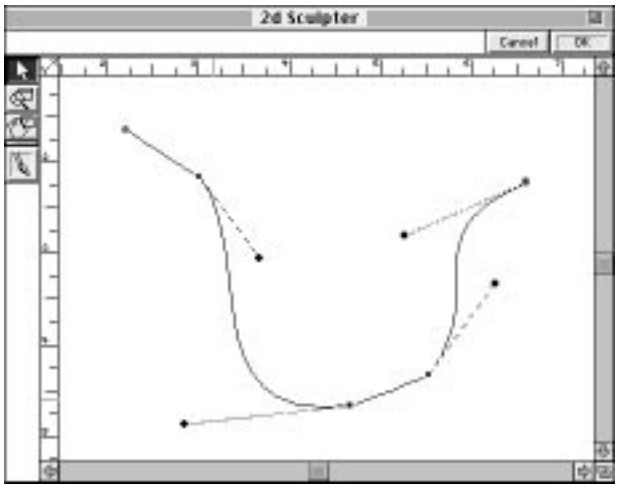
3. Select the Sphere tool from the Tool Palette; then click in the shape window to draw a sphere.
4. Close the Shape window.

5. Select the Open Polyline tool from the Tool Palette. Draw a line similar to the one below in the modeling window.



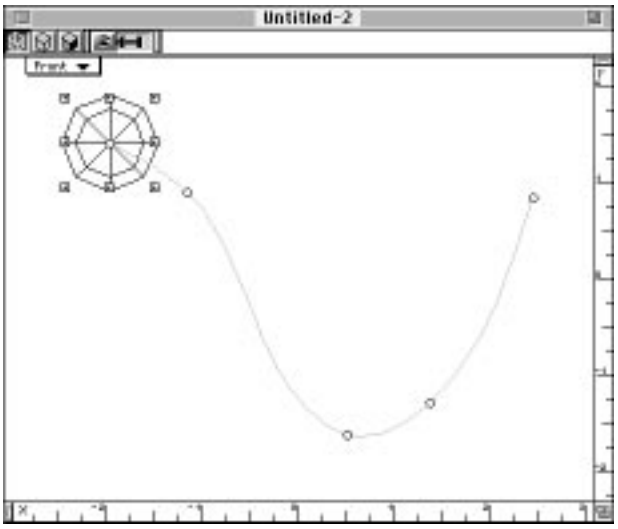
6. With the line selected, click on the 2D Sculpter button on the button bar. Highlight all of the vertex points by choosing Select All from the Edit menu. From the Options menu, select Convert to Bézier.

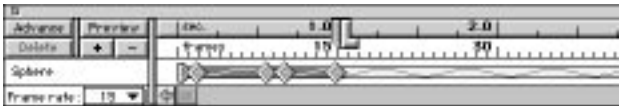
7. If necessary, you can make changes to the line by manipulating the handles on the vertex points. When you're satisfied with the line, click OK.



8. From the Path Filters sub-menu in the Animation menu, select Convert to Path. Note the start time for this example is set to zero; the duration is set to one second.

9. With Sphere selected, click the OK button. The Convert to Path extension will create as many markers as are required for a smooth animation.





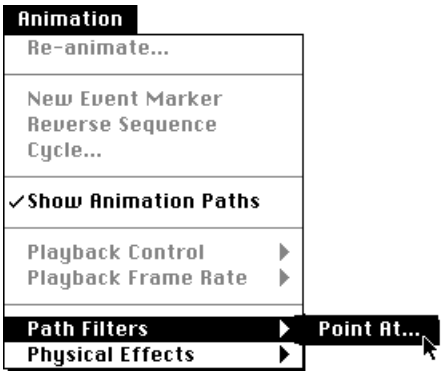
A CAVEAT

When you attach an object to an animation path, the Convert to Path extension will create on the path as many event markers as needed to provide for a smooth animation. Because these event markers are created automatically, it's possible to draw a line that produce an excessive number of event markers. When you draw lines intended for use as an animation path, do so with the intent of minimizing the number of event markers that will need to be created. It's generally a good idea to check the number of event markers created during a conversion before proceeding. That way, if your path has too many event markers, you can use the Undo command and try redrawing the line so it will generate fewer event markers.

Alternately, you can selectively delete any redundant event markers from the Animation palette after the path has been converted. With experience, you will be able to decide which technique is most efficient.

Point At

When present in the StrataVision 3d Extensions folder, the command to open the Point At extension appears in the Path Filters sub-menu of the Animation menu.



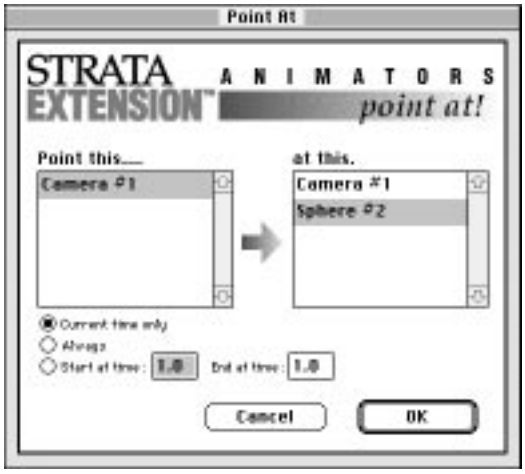
If the Point At command does not appear in the Animation menu, the extension is incorrectly installed.

DESCRIPTION

The Point At extension allows you to automatically point a camera object or local light source at an object within a Vision 3d model. Directional / global light sources cannot be used with this animation extension. The advantage this extension offers is obvious. You don't have to manually script a light source's or camera object's animation to make it point at an object…checking several views of the model in order to line up a camera or light source just right. Point At will do all the calculations for you! This extension also provides control over when and for how long the camera or light source is pointed at an object.

USING POINT AT

To use the Point At extension, you must have a at least one object and one camera object or light source in a model. The camera object or light source must be selected. Then, when you choose the Point At command from the Path Filter sub-menu, the Point At dialog box opens.



As you can see, the names of camera objects and light sources appear on the left side of the dialog, while the names of objects at which they can be pointed appear on the right. Notice cameras and light sources appear on both lists simultaneously.

Once an object from the left-side list and one from the right-side list have been selected, you can click on the OK button to complete the operation. If you try to point a camera or light source at itself, Vision 3d will prompt you to select another object at which to point the camera or light source. An object in your model can have more than one camera and / or light source pointing at it. But you can't point one camera or light source at more than one object.

If you have pointed a camera or light source at one object then decide to point it at a different object, simply select it again and open the Point At dialog to change the selection on the right-side list and click on the OK button. Alternately, you can choose the Re-Animate command in the Animation menu to change the Point At settings.

The time controls at the lower left of the dialog allow you to control when the camera or light source will point at the selected objects. By clicking on the Current Time Only radio button, you can point the camera or light source at an object at a specific moment in time (obviously, the current time as set by the time pointer). To point at an object over the entire course of an animation, click on the "Always" radio button. You can choose a specific portion of an animation during which to point at the selected object, simply by clicking the "Start of Time" radio button. For this option, enter a time value in that field and an ending time value in the "End of Time" field before clicking on the OK button.

This animation extension can also be used for non-animated objects. Simply point at the selected object at current time only. This is useful in projects where you only plan to render still images. Since you are rendering only one frame, that current time will always be time zero.

A CAVEAT

If you use this extension to point a camera object and/or light source at an object, they actually are not pointing at the object itself, but at the object's location in the 3-D space of your model. This is an important distinction. If the object in that location is moved to a new location, the animation path for the camera or light source is still pointing at the now-vacated position — an empty spot in 3-D space.

The Point At extension sets up cameras and light sources to follow an object as it moves on its animation path, but it does make them follow an object if the object is repositioned afterward.

Each time you move an object that has a camera or light source pointed at it, you must redo the animation path for the object pointing at it. This can be done with the Re-Animate command in the Animation menu.

Commands and controls used with animations

The following commands and interface controls are considered relevant to the animation features of the software. Most of the commands are found in the Animation menu. The list of commands is alphabetical, and an explanation is included with each item in this section.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or

add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in Chapter 1: General Operation.

Animation windows (rendering and play back)

As discussed in chapter 1, the window frame for each type of window used in Vision 3d contains the standard Macintosh interface elements. Those standard elements with which you should already be familiar include the title bar, size box, close box, zoom box, and scroll bars.

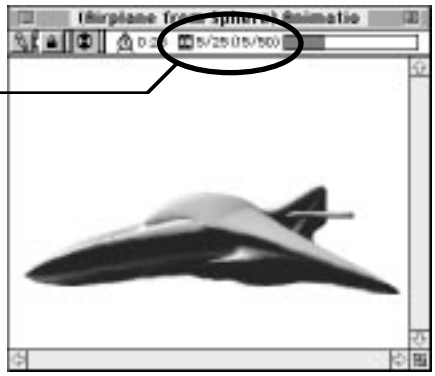
The specialized controls inside the windows are presented here. Mention is also made when the frame of the window types varies from the standard elements common to all windows on the Macintosh desktop.

When a rendering is initiated, Vision 3d opens a separate rendering window over the view selected. Both still images and animations use the same basic rendering window, with some additional animation information concerning the total number of frames to be rendered and the current frame being rendered.

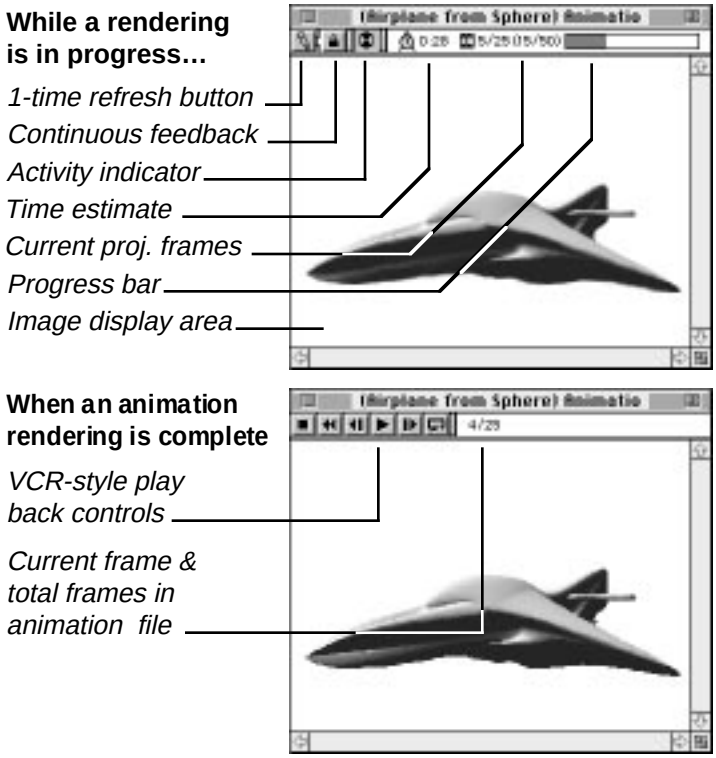
Rendering windows for animations include frame information

…In this sequence:

1. Current frame
2. Frames to render
3. Current frame within larger animation
4. Total frames of larger animation



A rendering window contains some specific controls that are unique to this type of window. These controls allow you to observe the rendering process as it occurs. Rendering windows serve as on-screen monitors during an image's creation. Note that the rendering actually occurs internally.

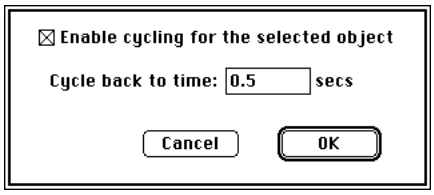


The main area of the window usually corresponds to the size of the rendered image. If the image size being rendered is larger than the screen, the rendering window controls will also include scroll bars and a re-size button to allow repositioning any part of the currently-rendering image within the window.

As with any desktop window, a rendering window can be moved around the screen or to the background, even while the rendering continues. To terminate the rendering, simply click on the window's close box.

Cycle command (Animation menu)

The cycle command allows you to repeat a sequence of event markers for the length of the animation. Cycle is available when an object is selected for which two or more event markers have been defined.



Choosing this command opens a dialog that contains a check box for enabling or disabling the cycle function and a field to designate the point on the time line at which the cycle is to repeat. When the last event marker is reached for the object, and the animation length is longer, the sequence is repeated from the time designated as often as needed to fill the balance of the animation length.

Cycle is useful for defining a repeating sequence that fills the entire animation without having to manually position event markers all along the time line. With this feature, you can set up a lead-in sequence, then cycle the main part of the animation over and over as needed.

New Event Marker command (Animation menu)

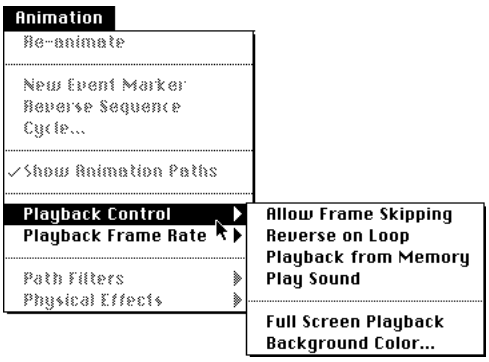
This command inserts an event marker at the current position of the time pointer on the time line. The New Event marker command is available when the time pointer is not positioned on an existing event marker.

The current state of the object at that moment on the time line is used for the event marker. It is a useful way to generate an “anchor” event marker from which the object’s movements or appearance can make sudden changes later in the time line. Anchor-type event markers are used to keep later event markers from causing the object to change before you want.

Playback Controls sub-menu (Animation menu)

Use the commands in this sub-menu to control the play back of animation files. Some of the commands apply only to PICS-

formatted files; some only to QuickTime Movies. As per interface guidelines, if a command is dimmed, it is not available in the current situation.



The Playback Controls menu item is active only when an animation file is open and its window is the active window. An explanation of the commands found in this sub-menu follows.

Allow Frame Skipping command

Use this command to enable or disable frame skipping in QuickTime Movies. The command toggles on and off each time it is selected in the sub-menu.

When frame skipping is allowed, the sound channel(s) that may be part of an movie file are given priority. The sound is then played at the correct speed and the frames are played as they can to keep up, sometimes requiring a frame to be skipped here and there to maintain the correct time sequence.

No sound is available when this command is disabled. If this command is turned off, so that all frames of a QuickTime animation are played in sequence — no matter how slowly — the Play Sound command is dimmed automatically.

Reverse on Loop command

The Reverse on Loop command allows you to alternate an animation sequence forward and backward when the loop button it pushed on the animation window. Reverse on Loop is available when an animation window is open.

The Reverse on Loop command is a check item that toggles on and off each time it is selected from the menu. When enabled, a check mark is visible in front of it as the menu is pulled down. If

more than one animation window is open at one time, this option can be set independently for each file.

When enabled, an animation sequence will play forward first, then reverse the frame sequence upon reaching the end so that the animation plays backwards. When the first frame is reached, the sequence reverses again to play forward. The cycle is repeated until the stop button is pressed on the window, the loop button is turned off, or this command is toggled off.

If the loop button is on, but this option is disabled, the animation will play the frame sequence to the end, then begin again at the first frame, repeating the cycle until the stop or loop buttons interrupt it.

PLAYBACK FROM MEMORY COMMAND

The Playback from Memory command allows you to load the frames of an animation into memory. Playback from Memory is available when a PICS animation file is open in a viewing window. Playing a PICS animation from memory is more desirable than playing it from hard disk because the animation can then play at its true speed, thus producing a more pleasing appearance …and a more accurate representation of its creator's object motion and timing.

Normally, only one frame of a PICS animation is in memory at a time. In this mode, each frame must be spooled through the window from the hard disk, causing the animation to play slower than it should and without any "flow" in the motion.

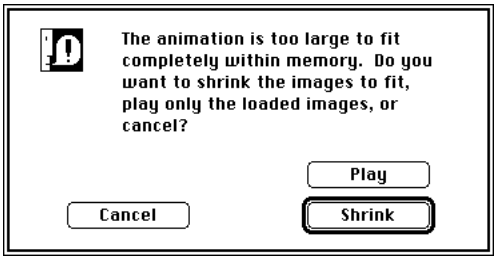
The Playback from Memory command is a check item that toggles on and off each time it is selected from the menu. When enabled, the command has a check mark in front of it as the menu is pulled down. If more than one animation window is open at one time, this option can be set independently for each file.

The color depth of the monitor on which the animation is to be played determines the color depth of the images as they are loaded into memory, i.e. — an animation playing on a monitor set to 256 colors (8-bits per pixel) will take less room in memory than one set to

16.7 million colors (32-bits per pixel). Setting a monitor to a different color depth after an animation has been loaded into memory will cause Vision 3d to re-load the images at the new bit depth. The color depth of the frames in the file stored on hard disk is unaffected.

Turning this option off for a window in which the animation has been loaded into memory purges the buffer allocated for the frames; playback for that window then reverts to spooling the file from disk.

Selecting this command causes Vision 3d to begin loading all the frames into a buffer in memory. The size of that buffer depends on the amount of free memory available. Under System 6.0, the memory comes from within the partition allocated for Vision 3d. You can improve the probability there will be enough memory by specifying a larger partition in the application's Get Info dialog while on the desktop. Under System 7.0, the buffer is allocated from available memory outside the partition.


The animation is too large to fit completely within memory. Do you want to shrink the images to fit, play only the loaded images, or cancel?
Play
Cancel
Shrink

If during the loading operation, there is not enough room to fit all the frames of the animation into memory, Vision 3d gives you the option of reducing the frame size so they can all fit or to just proceed with playing the partial animation that will fit at full size into memory.

Choosing the shrink-to-fit option restarts the loading process so the frames can be downsized as they come in. Choosing the play option allows you to view that portion of the animation that was able to fit, but no more. Choosing cancel terminates the loading process and turns off the Playback from Memory command. Further viewing of the PICS animation reverts to reading the frames from the hard disk one at a time.

PLAY SOUND COMMAND

Use this command to turn sound on or off when playing a QuickTime Movie file. The Play sound command is a check item that

toggles on and off each time it is selected from the menu. When enabled, the command has a check mark next to it in the sub-menu. If more than one movie file with sound is open, only the animation in the active window plays its sound. This command may be set independently for each animation as its window is made the active window.

If the animation open in the active window is a PICS file or a Movie file without sound, the command is dimmed.

If the Allow Frame Skipping command is disabled so that Vision 3d is constrained to play every frame of an animation, this command is dimmed and Vision 3d will not play the sound even if present in the animation.

FULL-SCREEN PLAYBACK COMMAND

This command provides the option for blanking the desktop behind the open animation window. The “Full Screen Playback” command is a check item. It toggles on or off each time it is selected.

Note that the size of the image frame does not increase to fill the monitor’s screen, the normal desktop is just temporarily hidden by a solid color outside the animation’s frame.

When full-screen playback is on, the menu bar is invisible. Menus are fully active, however, and can still be pulled down to choose commands. You can return to normal desktop operation by turning off full-screen playback through the Animation menu or by simply pushing the Escape key.

Keyboard Equivalents for Animation Window Controls.

Keyboard controls have been added to make the operation of the Animation window easier, especially in full-screen playback mode. The equivalents are:

Start playing animation	UP arrow, RETURN / ENTER key
Stop playing animation	UP arrow (again) RETURN / ENTER key (again)
Next frame	RIGHT arrow
Previous frame	LEFT arrow
First frame (rewind)	DOWN arrow
Turn on looping	BACKSLASH key (\)

You can also simply click on the image with the mouse to start and stop the animation. If the full-screen playback is turned off — so that the controls at the top of an animation window are visible — you will also see those buttons echo the keys as they are pushed.

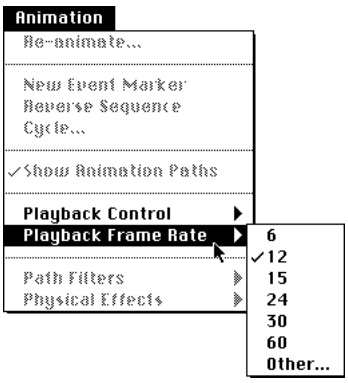
BACKGROUND COLOR COMMAND

This command sets the color of a background that may be used behind an animation while it is playing in full screen mode. The default color is black.

The Background Color command opens the Apple Color Picker dialog box to select a color.

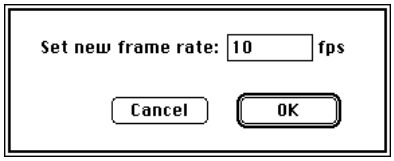
Playback Frame Rate command (Animation menu)

The Playback Frame Rate allows you to set the speed at which an animation will play. This command is available when an animation window is open and is the active window on the desktop.



When an animation file created by Vision 3d is opened, the frames per second rate set in the Animation palette when it was rendered is automatically pre-selected in this sub-menu. You can use that setting, choose to use a different playback rate from the sub-menu, or set a custom frame rate through the Other option.

Choosing “Other” from the sub-menu opens a dialog in which you can enter the frame rate you wish for playback. A value between one and 60 can be entered.



You should know that the maximum speed at which the animation window can play an animation depends on the capacity of the hardware. For a PICS file to play at the speed specified by this command, it must be loaded into memory.

On the other hand, a QuickTime Movie file may be able to play in real time from hard disk, depending on the compression method used, the size of the frame on the screen, and if the hardware is capable of spooling the image data fast enough from the hard disk through the SCSI port and into video RAM. Some of the newer high-end Macintoshes are capable of this performance.

If a PICS animation is being played from hard disk, it will play as fast as the images can be read and displayed on the screen, without the benefit of file compression or spooling. It will be substantially slower than the target rate set here, no matter how capable the hardware.

Even when playing from memory, depending on the frame size, color depth, and speed of the CPU, few factory-standard Macintosh computers are able to display the frames fast enough to reach a true 60 fps play rate, which is the maximum allowed by the software. If the play rate chosen is faster than the hardware's ability to respond, the animation will be played as fast as possible, whatever that speed may be.

Re-animate command (Animation menu)

Use this command to re-calculate the most recent animation effect that has been applied to an object. This is useful if you wish to change the settings for a special effect provided by one of the animation extensions available for Vision 3d.

If no animation effect has been applied to the selected object, the command is dimmed. If more than one object is selected, the command is also dimmed. It works on only one object at a time.

This command is specifically intended for use with the animation extensions installed with the software. The available

animation extensions are listed in sub-menus at the bottom of the Animation menu.

Note that re-animating an object often requires more memory than animating it the first time. That's because the first set of event markers and the animation path generated by an animation extension are kept intact during the re-animate operation. If a re-animation effort is successful, the first animation set is deleted. If the re-animation fails for some reason, the first animation set is restored to the model

Reverse Sequence command (Animation menu)

The Reverse Sequence command allows you to reverse the sequence of a selection of event markers along the time line. Reverse Sequence is available when two or more event markers are selected. They do not have to be contiguous.

Selected event markers are physically moved on the time line to their new locations. The reversal occurs symmetrically around the logical center of the selected event markers. The selected event markers do not have to be contiguously grouped. Un-selected event markers within the time span being reversed are left stationary.

Show/Hide Animation Palette commands (Windows menu)

The Show Animation Palette command displays the Animation palette. This command is available when the Animation palette is hidden. Hide Animation Palette is available when the Animation palette is showing. Choosing the Hide Animation Palette command is the same as clicking on the close box of the Animation palette.

Alternately, you can use the show / hide button for the Animation palette found on the Button bar.

Vision 3d displays only one Animation palette, even if multiple model documents are open. However, when the active model changes, the palette is redrawn to reflect the status of the current model. This and all the other palettes in Vision 3d always manage the active document.

The Animation palette's show / hide status when a new session is started is determined by the previous session. If the palette was showing when the application was quit the last session, it will be showing when a model is opened in the current session. If it was hidden at quitting last time, it will be hidden on startup this time…the status is retained from session to session.

Show Animation Paths command (Animation menu)

Use this command to show or hide the animation paths scripted for objects. When enabled, the animation paths of objects may become visible when the object is selected. This command toggles on or off each time it is highlighted in the Animation menu. When on, a checkmark is visible next to the command on the menu.

Note that you must turn this feature off to allow the moving of object origin points on objects. When this command is toggled on, the position of ALL object origin points is locked relative to their respective objects.

CHAPTER 8:
RENDERING IMAGES

The effort invested in creating a model — modeling the objects, applying textures, assembling a scene, and positioning lighting — inevitably brings you to that anticipated moment when you are ready to render the finished image(s). When done correctly, the result can be a very satisfying moment. As you receive accolades from your peers for the incredibly realistic still image or the animation that breaks new cinematic ground, be sure not to tell them how easy it was (…in some ways, yet hard in others).

When you click or drag the camera tool in a view, the "snapshot" event initiates a rendering and brings all the rendering prowess of Vision 3d to bear on the model you have created. The rendering process is the electronic counterpart to developing the film in a real camera.

Obviously, the approach you take to set up the finished rendering varies with the intended use of the image or animation. If the rendered image or animation will be viewed solely on the computer screen, the resolution and size may not be as critical as when the image is to be printed in a high quality, full-color sales piece or the animation is to be part of an industry video distributed to thousands of customers.

Note that no matter what its ultimate use, the basic steps for setting up and rendering your masterpiece are the same…only the size and resolution settings change.

Vision 3d has ten different algorithms or methods available to render your images — from wireframe to RayPainting™. All the renderers are built in (at no extra cost), thus they are readily available for previews during modeling, scene construction and fine-tuning adjustments, as well as for the finished renderings. Notice that Vision's RayPainting and RenderScript renderers are trademarked.

The type of rendering method used determines the effects that will be seen in the rendered image. As more effects are requested (through the Rendering Options command in the Rendering menu), the quality of the image is enhanced. However, there is a tradeoff. More rendering effects usually mean longer rendering times. There are ways to minimize the impact as you add more photo realism. This chapter will address the options and suggest ways to improve rendering performance. You should realize there is no magic setting and no "free lunch." Performance

improvements will come by intelligently picking the options, by carefully monitoring memory partitions and other internal settings, and by working within realistic expectations. If you do this, you will find Vision 3d a very capable package that delivers on its promise of 3-D visualization.

The actual time required to render an image or animation depends on many variables. Some of the variables directly affecting rendering efficiency are:

- **Platform and hardware configuration. The primary factor in rendering speed is** raw CPU "horsepower." Because Vision 3d is so CPU intensive, it is obvious that faster Macintosh platforms will render in a shorter period of time than lower performance versions. Hardware acceleration is probably the single most significant improvement that can be made to a computer for 3-D rendering purposes.
- **System software version and configuration.** Vision 3d requires version 6.0.5 or later of System software. Later versions of the Mac OS have improved efficiencies and use improved editions of Color QuickDraw, QuickTime, and memory management. System configurations like the RAM cache size, MultiFinder, and the number of active inits/extensions also affect overall speed.
- **Amount of total memory and the partition size given Vision 3d in MultiFinder.** The amount of RAM available to Vision 3d determines how large of an image segment it can render at a time. Less memory means spooling for rendering may have to be used. It also means smaller buffers and smaller segments for rendering, requiring more disk access time to swap segments in an out of the virtual image cache on the hard disk.

  HOWEVER, it is not always best to allocate all the available RAM to Vision 3d. The general rule is that you should allocate only enough to comfortably hold the model. The appearance of the memory chip in a rendering window indicates that Vision 3d would like more memory for its partition, but you should increase the partition size in conservative amounts rather than large chunks. If you have allocated all available RAM to Vision 3d, and the memory chip still appears, then you must either: 1) reduce the quality of the rendering, 2) use Phong instead of raytracing for the rendering method, or 3) install more RAM.

  If Vision 3d needs more memory during rendering, and it is available, it can get some from that available outside its partition if

you are running System 7.0 or later. (System 6 does not have this feature.) Some operations must be within the partition (thus, the memory chip appears when this is not sufficient for those operations), but other things can be done in this additional memory provided by the system. The other things includes buffers for texture maps, pre-loading the next segment for rendering, etc. This allows Vision 3d to reserve the RAM inside its partition for the actual rendering process. If you allocate all the available RAM to Vision 3d, thinking that it will run better, you may see an "Out of Memory" alert instead, even though there should be sufficient memory.

- **Options chosen in Rendering Effects dialog.** Enabling a rendering effect means that the renderer must perform the calculations to include that effect in the image, even if no surface exists in the model using the effect. Disabling an effect allows the chosen renderer to ignore that effect, even if surfaces are defined that would use the effect. Reflectivity is a good example: Even if there are reflective surfaces in the model, when reflectivity is not enabled as an option for an image, the renderer can pass over that part of its code; saving that much time in the process.
- **Size and resolution of the image to be rendered.** The rendering time is directly related to the number of pixels in the image. The number of pixels is mathematically determined by the image's dimensions and the dots per inch.
- **Type of objects in the model.** As discussed in other chapters, primitives use less memory and render faster than polygonal or spline objects. Of the various primitive shapes, spheres render the fastest of all.
- **Number of objects/polygons in the model.** Models with many objects require more overhead to keep track of them. Vision 3d must make several decisions while rendering an image. One crucial decision is which surfaces are visible from the chosen viewing position and which are not. Another factor is which objects are in front and which are behind. If the objects are described using polygons or a polygonal mesh rather than primitives, the time required to sort through a model's internal database takes even longer.
- **Number and types of light sources in the scene.** The calculations performed on illumination increases exponentially with each additional light source. Because you can use an unlimited number of light sources, the potential for abuse is very high. Each light source must be accounted for. Spot lights have additional calculations

related to their cone angle and soft edges at the cone of light boundary.

- **Types of textures used for object surfaces.** Solid textures generally render faster than surface textures. Reflective and transparent textures add rendering time, especially when using raytracing for refraction. Textures using surface maps require additional buffers for the maps so the renderers can refer to them during the rendering process. Each surface map increases the model overhead.

Some of these factors can be altered or optimized to yield better performance for Vision 3d without adversely affecting the image quality. They are presented here so you will be aware that you can improve performance by turning off options that are not actually required or removing objects that make no contribution to the image. This chapter can make you aware of the things that most directly affect rendering speed and you can make more educated choices in the modeling and rendering processes. On the other hand, if you actually want to include the items that will make a rendering take longer to complete, you can also be prepared to invest the time required to obtain the worthy results you desire.

Of course, you are encouraged to use as many rendering options as needed to obtain the desired results. The incredible range of options available in this software is what makes Vision 3d the superior product and photo-realistic standard that it is in the industry.

Much of this manual has focused on how to create objects, define textures, insert lights, etc.…then assemble them all into an intelligible scene. This chapter now addresses how to render that scene into an intelligent image. There are many options controlling the rendering process. Rendering is a simple process when using the standard settings provided in the software. But, that's only the beginning. Because of its many photo realistic options, Vision 3d also offers a lot of power to the advanced user willing to try them.

There are three areas requiring attention when rendering an image from a three-dimensional scene:

- **What is the content of the image as determined by the camera position and focal length?** The camera is designed to imitate an actual camera and is the mechanism used to gather the 3-D scene into the 2-D image.
- **How will the pixels that represent the image be decided?** In other words, what rendering method will be used? The 2-D image to be rendered is defined by the rendering window. It is the frame of the

image. The image created in the rendering window is analogous to photographic film in the process of developing.

- **The image produced must be accurate.** How good is the renderer? Can it, in fact, do what you are asking of it? Does it faithfully capture the image you intended, or does it look obviously computer-generated? What can you do to achieve the "look" you're after?

A digital "pinhole" camera

The purpose of a camera is to obtain an image (2-D, of course) from a three-dimensional scene. Whether this is done in the physical world by clicking the shutter on a real camera or in the computer by clicking the mouse button, the steps are actually quite similar.

Instead of actual objects, lights and cameras, the rendering algorithms in Vision 3d work on internal databases of non-existent objects created and assembled by you in 3-D space to resemble (or represent) something recognizable. The presence of light in the model, its interaction with surfaces, and its reflection toward the viewing position are simulated by algorithms that operate on this internal data. The principal job of the Vision 3d interface is to let you communicate enough information to form this replica, i.e. — object geometry (including scale and proportion), object position, surface descriptions, placement of light(s) as well as the camera's position, so it can be captured on the view plane used by the camera and rendered with total integrity to the model.

There are four factors that determine what a rendered image will look like:

- **Position.** In the real world, no matter how automatic a camera is, someone has to decide where to put it…
- **Orientation.** …and point it at something…
- **Field of View/Focal length.** …then decide how much of the world or scene is visible in the camera lens…
- **Resolution.** …and, of course, the type of film determines the quality of the image.

The metaphor comparing the process of rendering computer-generated images and taking snapshots in the physical world has been referred to throughout this Users Manual. At some point, however, the metaphor breaks down. Because of hardware limitations, 3-D modeling in the computer still requires some accommodation to the digital world represented in the model.

Better than a real pinhole camera

The camera design used by Vision 3d is a simple pinhole camera. This camera concept was chosen because it is the most computationally efficient. With the pinhole camera, the pixel color at any given point on the viewing plane (or vice versa, in the rendered image) can be determined by projecting a straight line from that point through the pinhole aperture and out into the model.

There are several advantages to simulating a pinhole camera on the computer. The most obvious enhancement to a real pinhole camera is an infinitely small aperture, which real cameras cannot match. Physical cameras have a measurably larger aperture that can focus objects within only a narrow range of distances. The extent of that focus range — known as the "depth of field" — is related to the diameter of the lens aperture, sometimes referred to as the "f-stop." The larger the opening, the narrower the depth of field. So, while physical cameras have limited ability to resolve detail with distance, by contrast the computer camera has perfect focus at all distances.

Greater focal range

Vision's version of the pinhole camera also provides for both orthographic and perspective projection. Hence, the adjustable focal length controls on both the modeling views windows and camera object window have a much greater range than can be achieved in the real world.

In essence, orthographic projection uses an infinitely long telephoto lens to achieve its display on the view plane. The stronger the telephoto effect, the more foreshortening is apparent in the image; causing all objects of similar scale to appear the same size regardless of their distance from the camera. On the other hand, a perspective lens setting produces images in which objects near the viewing position appear larger than objects positioned farther away. Vision's extreme wide angle is likewise beyond the physical ability of physical lenses. The amount of perspective in a view is inversely proportional to the focal length and directly proportional to the field of view.

Chasing after light rays

Another significant difference that exists in Vision 3d, when compared to the physical world, is the approach taken for dealing with

light in a scene. Consider for a moment how your brain processes images of the world around you. As you look around a room to form a visual image, you are sub-consciously asking yourself, "Where is the light going?" In the physical world, objects are awash in light from multiple light sources or reflecting from other objects, with only some of it actually striking the retina of your eyes where you are standing.

The computer must take a more conservative approach when it analyzes how images should be formed. A much more efficient approach is to interrogate only the light rays striking the viewing plane and trace them back through the model to their sources. All other rays of light produced by those sources must be completely ignored. The computer must ask, "Where did the light come from?" Notice the difference between you and a computer in approaching the same task? The computer cannot afford (and does not have the power) to spend time looking for and digesting light rays that never reach the camera's lens. You can easily do this several times a second with your brain's superior abilities, but the computer is too limited. Thus, for all its abilities, the computer comes in poor second to your brain. When you judge the level of photo-realism in a rendering, you are actually comparing the computer-generated image to what your brain can create with the same information.

Shading objects

The rendering process calculates the appearance of objects in a scene in a specific lighting situation. It does this by calculating the nature of the light reflecting from a surface as a function of the light striking it. There are three main factors that must be factored into the equations used by the renderers in deciding how object surfaces should appear. These are presented on the next page.

- **Color.** The interaction of the surface color with the light color is critical to the appearance of objects. The color of most real objects usually changes from point to point on their surfaces. To appear equally "real" in Vision 3d, the renderers must be able to support complex texture mapping to reproduce that level of object detail.
- **Orientation.** The position and orientation of an object with respect to the light sources must also be considered. The greater the distance, or the more a surface faces away from the light, the darker the surface appears. On a curved object, the surface's orientation changes from point to point. For some materials, even the color

changes with orientation. This is true especially in areas of high specular reflectivity called “hot spots.”

- **Smoothness.** The relative roughness or smoothness of a surface determines how light is reflected from it. A surface where the light’s angle of incidence approximately equals the angle of reflection will appear shiny. In exact situations, the object acts like a mirror. A surface where the light is scattered at random upon reflection will appear dull.

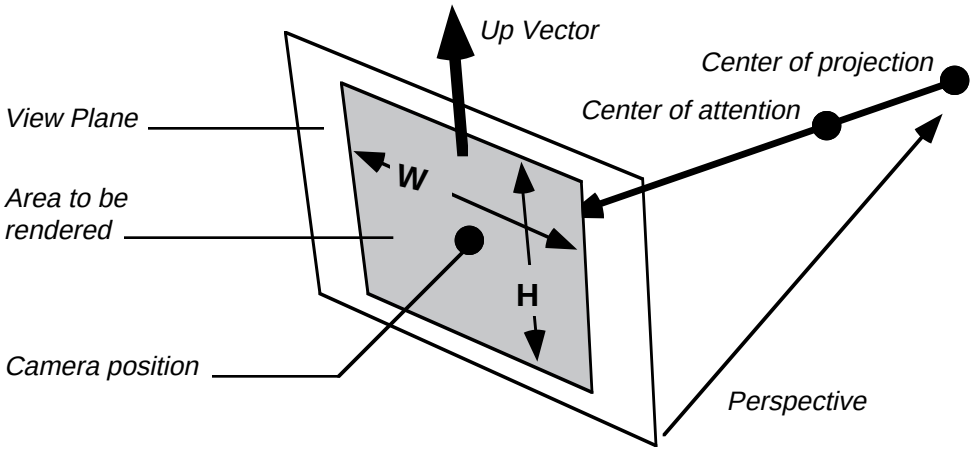
The permutations of combining these factors are as endless as the surfaces they can represent. The appearance of the light reflecting from a surface carries important visual information about its composition. Often texture maps are the only way to represent surface variations too small or complex to be geometrically modeled. Determining the reflected appearance — that is, the direction and color of the light coming from a surface — can be complex and computationally expensive; hence the occasional extended rendering times. When done right, however, the results are rewarding…and very realistic.

The rendering methods used, the nuances of color and shading, the setting of each pixel in the context of the overall image — all these things exert powerful, yet subtle influences over the final appearance of an image. Time invested in fine-tuning texture settings, whether that takes the form of editing the fields and surface maps in the actual texture-editing dialogs, or in their application to the objects through mapping style, orientation, and scale application, probably offers more opportunity for creativity and innovation than any other aspect of the modeling / rendering process.

Perspective

The viewing position apparent in the render image(s) depends on the active view used by the camera tool when taking the snapshot.

While you may use any open view for a viewing position, camera windows usually lend themselves to this task better than modeling windows.



Vision's method of specifying the viewing position is similar to the way an actual photographer might think about positioning his camera. Seven parameters are obtained at the moment the camera tool is clicked in a view:

1. **The center of projection of the view used by the camera** (this corresponds to the location of its view set center in model space).
2. **The direction the camera is aimed** (the view's position in model space).
3. **The orientation of the camera** (which way is "up?").
4. **The distance from the objects** (determined by the view plane's distance from its view set center and that view set's position in space).
5. **The width of the area to be rendered** (determined by the view's width in the case of a single click in the view or the portion of the view defined by the drag marquee if the camera tool is dragged in the view).
6. **The height of the area to be rendered** (using the same data as in #5).
7. **The focal length of the view** (its view angle and resultant perspective).

These parameters are made available to the renderer chosen to perform the rendering task. The data is then plugged into the rendering

calculations; allowing the renderer to faithfully reproduce the viewing position chosen for the rendered image.

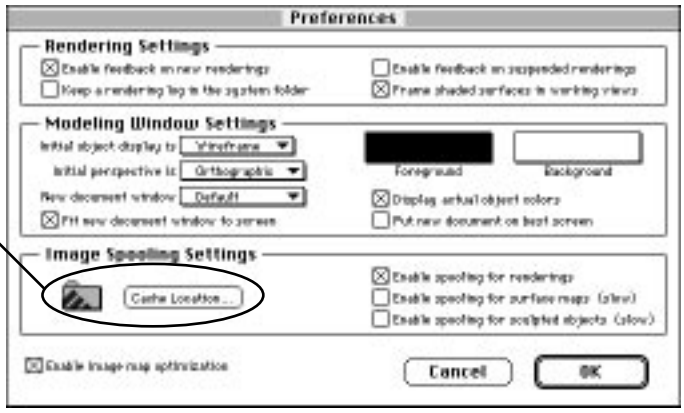
Internal image handling

When a rendering is initiated, the representation you see on the screen as it is rendering is just for your benefit. The image is rendered and stored internally in full 32-bit color or 8-bit grayscale, independently of the image displayed in the rendering window.

If spooling for rendering is enabled, Vision 3d sets up a virtual image cache on your hard disk at the location designated in the Preferences dialog. You may locate the cache on any mounted volume that has sufficient free space available. This is where the rendered image is assembled as it is being rendered.

Set the virtual image cache location with this button in the Preferences dialog

...for use during renderings when the "Enable spooling for renderings" checkbox is checked.



In addition, one or more buffers are opened in RAM to handle the various components of a rendering, i.e. — surface maps, depth sort of visible surfaces, light sources and their illuminations, etc. The size of these buffers depends on the following factors:

- **The actual algorithm performing the rendering.** Raytracing and RayPainting typically require more memory than other rendering methods.
- **The size of the rendering.** Larger images contain more detail…and more pixels, thus requiring more memory to store.
- **The model's complexity.** Each surface map requires one or more buffers for the renderer to reference.

- **The amount of memory available.** The more memory available (in and outside of the application), the faster it will render.

If the entire rendering can be handled in memory (buffers and all), no image cache is created. Tight memory constrains the size of the rendering buffers, relying more on the virtual image cache. If sufficient RAM is not available, Vision 3d employs the virtual image cache to store and retrieve data on a hard drive during the rendering process. This method allows the image-creation process to continue.

Rendering in the background

One of the most powerful features of Vision 3d is its ability to render images in the background. Any number of renderings can be initiated at the same time — within the limits of available memory — each in a separate rendering window. The only limitation is that no alteration can be made to the model data during rendering.

If any model modifications are attempted while a rendering is in progress, an alert dialog is displayed warning of the consequences if the change is allowed — which is to terminate the current rendering, since the data upon which it is based would no longer be valid.

Background rendering frees the desktop's foreground environment to work on other tasks. These tasks may include using desk accessories, setting up views for other renderings or, if in System 6 MultiFinder or System 7, running another application.

You should realize, however, that rendering in the background splits available CPU time among the rendering(s) in progress and the activity in the foreground. The foreground activity ALWAYS takes priority, but there are many CPU cycles that go unused. So, while background rendering speed can be somewhat slower, it is better than none at all. If a particular rendering window is made the active window by bringing it to the foreground, it takes priority and all processor cycles will be given to that rendering until it is completed or until another window is brought forward.

Of course, it doesn't take a rocket scientist to figure out that the more images the computer is rendering at one time, the longer it takes to complete each one. The performance loss is even higher than one might expect because of machine cycles lost while switching between competing renderings. In multiple rendering situations, it is often more productive to suspend the renderings, then queue them up to render one at a time overnight or at other non-use periods.

Suspending and restarting renderings

Vision 3d has the option of suspending and restarting renderings as often as needed to fit your schedule. If you need to work on some other project that requires the total computer, you can suspend and save a Vision 3d rendering at its current state of completion, then restart it later at the point where it left off.

In addition to the project flexibility this feature can give you, two more benefits of using suspended renderings are presented on the following page.

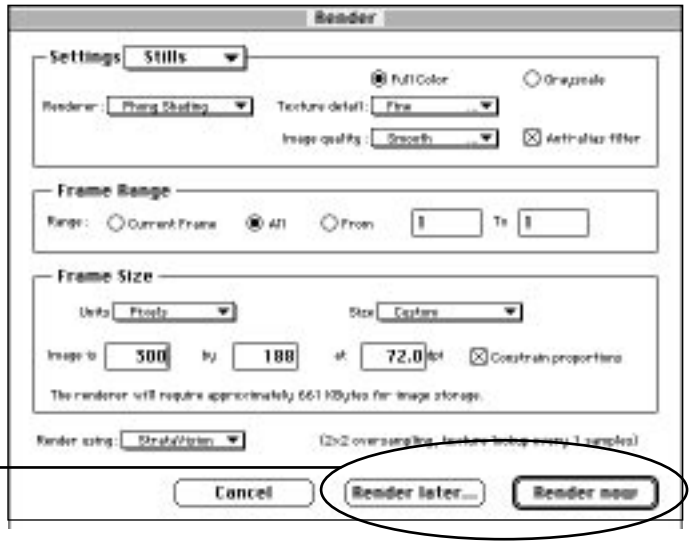
- **The original model does not have to be open to restart a suspended rendering.** Because the file is self-contained, suspended renderings tend to be a little more memory efficient. You can also restart a suspended rendering on another machine without having to provide the original model.

  NOTE: *If the model you are rendering uses solid textures when moving a suspended rendering from one machine to another, you must ensure the appropriate texture extensions are present on the second machine. Otherwise, the suspended rendering cannot open. This applies to model files as well. Please refer to "Chapter 5: Using Textures" for more information. Missing texture extensions are not usually a problem, but an incomplete installation of StrataVision 3d can become problematic. You should be aware of this aspect of transferring files between machines on which Vision 3d is installed.*

- **Restarted renderings automatically save upon completion; replacing the suspended rendering file with a finished image.** (That means, if a suspended rendering completes at 2:00 am in the morning and automatically saves, you won't lose all your work if an unplanned power outage occurs at 2:15 am!)

  To suspend a rendering at the time it is initiated:

1. **Hold down the Shift key while using the Camera tool.** Clicking once in a view takes the entire view as the snapshot to be rendered. If you do not want to render the entire active view, you can use the Camera tool to drag a cropping marquee in the active view. The marquee defines the extent by cropping in on the scene in the active view.
   Alternately, you can directly choose the Render command in the Rendering window. By design, this also defines the snapshot to be rendered as the entire active view.
2. **The Render dialog appears.** This dialog allows you to set the rendering parameters prior to beginning the actual rendering process. Note that the top section of the Render dialog is similar to the Rendering Options dialog. You can use a pre-set from the pull-down menu at the top of the dialog, or choose settings directly. The Render dialog gives you the option of beginning the rendering immediately or deferring the rendering until a later time. Of course, you can also cancel this process, in which case no rendering is initiated and you are returned to the model to do further work.



Render dialog has the option to render now or later

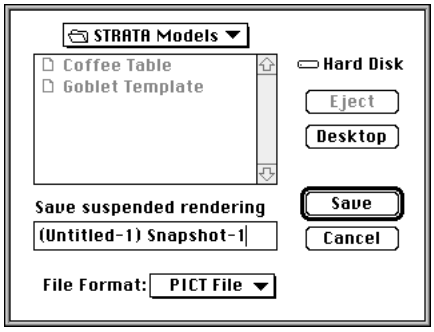
If you choose to render later, a directory dialog appears asking for the name of the suspended rendering file that will be saved and where you want to store it. You are free to choose a location on any mounted volume. Assuming you have write privileges to the chosen volume, a suspended rendering file is saved containing the model data and the rendering parameters requested. No rendering work has yet been performed on it, but it may be opened at your convenience to begin the rendering.

To suspend a rendering already in progress:

1. **Make the rendering window the active window.** If the window is not the active window, but is visible behind another window, just click the cursor anywhere in the window to bring it forward. If the window is completely hidden, use the Snapshot Windows sub-menu to select and make it the active window.
2. **Choose the Suspend Rendering command from the Rendering menu.** You are asked for a name and location to save the file.

Choosing the Suspend Rendering command normally will save the rendering project in its current state of completion and close the rendering window, thus terminating the project (for now). Alternately, you can hold down the Option key while choosing the Suspend Rendering command. This saves the rendering in suspended rendering form (in its current state of completion), but will not terminate the project. The rendering will continue on after the suspended rendering file is saved. This feature may be used to create a backup of your rendering project, so that if anything untoward should happen before completion, such as a power failure or system crash, only the work performed since the last save will be lost.

3. **Click "OK" in the Save As dialog box.** The rendering is saved with all the data required to restart it at some later time.



Another opportunity for suspending a rendering in progress is presented if you just close a rendering window. If the rendering has been going for more than four minutes, an alert appears asking if you want to suspend the rendering or just terminate it.

Clicking on the Yes button allows you to save a suspended rendering in the same manner as using the Suspend Rendering command does. Clicking on the No button closes the rendering window and throws away the work completed. If that rendering was from a previously-suspended rendering, the file is left untouched. Clicking on the Cancel button ignores the close attempt and continues working on the rendering.

Playing back partially-completed animations

The single frame of a still image is rendered to completion then saved, and the project is done. But an animation rendering is more complex. Vision 3d must save each frame of the animation as it is rendered before proceeding to the next. The frames of an animation are appended to the end of the file as they render. If you stop an animation before completion, AND DO NOT SUSPEND IT AGAIN, you will be

able to play the animation as far as it is done. The problem is you can't restart it, because it is no longer a suspended rendering.

If you want to be able to continue rendering an animation, you must suspend it. THIS IS IMPORTANT: When a suspended animation rendering is restarted and it renders and save one more frame, the suspended rendering shell is stripped off the file. If you quit an animation in the process of rendering without suspending it, that is the end of the project. You must suspend it to be able to continue rendering when you start a new session. But here's the catch: Once suspended, the frames of an animation are not available for playback.

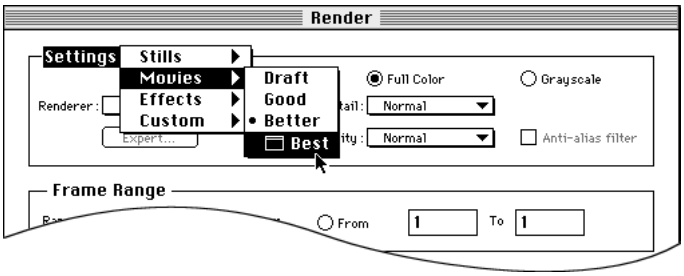
So, how do you play back a partially-completed animation without ruining it for further rendering? It is a simple matter of returning to the Finder / Desktop and making a duplicate of the in-progress animation file while it is not a suspended rendering. Then bring Vision 3d forward again and suspend the rendering. Now you are free to open the copy of the animation file and preview the frames that are completed to this point. If satisfied, re-open the suspended rendering file and continue on with the project.

Setting and using preset rendering options

The Preferences file residing in the System Folder contains some preset groupings of rendering options entered at "the factory." These settings are given names indicating their most appropriate usage, and are available for all models.

Use the pull-down menu on the Tool palette

…Or the pull-down menu in the Render dialog box



You may access the pre-defined rendering settings through a pull-down menu at the bottom of the Tool palette or through the Render

command in the Rendering menu. The Render dialog contains a similar pull-down menu, as shown in the illustration on the previous page.

Choosing a rendering options pre-set from the pull-down menu on the Tool palette automatically selects the Camera tool. If you select a pre-set from the Movies sub-menu on that menu, the Camera tool changes to a movie reel to indicate a set of rendering options have been selected for rendering animations. If you select a preset from the Effects sub-menu, the Camera tool changes to a paintbrush to indicate one of the RayPainting™ effects has been chosen.

Camera tool's three icons

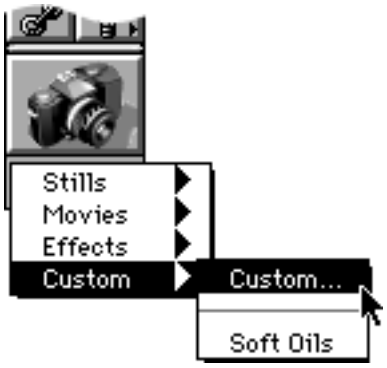
Stills *Movies* *Effects*

Choosing a pre-set rendering option causes a "bullet" dot to appear next to that preset. The dot remains until another pre-set is chosen, or unless the settings are manually altered in either the Rendering Options or Render dialogs. If more than one of the pre-sets happens to match the currently-selected rendering options, dots appear next to all that match. Thus, it is possible that the "Movies•Better" can be the same collection of rendering settings also used by "Stills•Better," or one or more custom settings you configure and save.

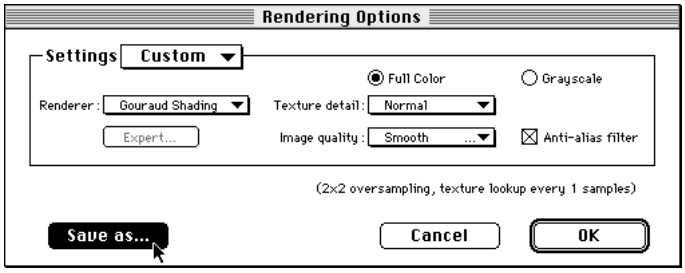
To switch to another preset rendering option, simply click on this pull-down menu and select the desired entry. The factory pre-sets cannot be altered or deleted from the lists in the pull-down menu. However, the rendering options you define and save can be deleted (see next topic).

Saving rendering options by name

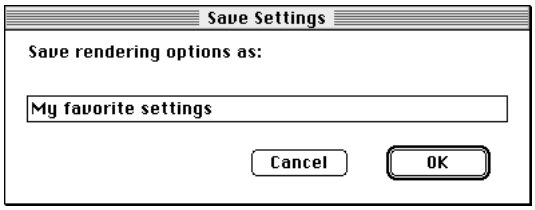
If you assemble a particular set of rendering options that you would like to recall later — either in the current project or in another model — you may save them by name and attach that name to the list of pre-sets available in the rendering options pull-down menu.



The Rendering Options dialog box is opened by choosing the Custom entry in the pull-down menu on the Tool palette, or by choosing the Rendering Options command from the Render menu.



Saving custom rendering options is done through the Save button in the Rendering Options dialog box. You will be prompted for a unique name to store setting under. Clicking “OK” adds the name to the list in the Set Rendering sub-menu.



NOTE*: Do not confuse the Rendering Options command with the Render command, both of which are in the Rendering menu. It is true that they have some common elements in their respective dialogs, so at first glance you might think they do the same thing. Upon further investigation, you will find that the purpose of the Rendering Options dialog is to set (and optionally save) rendering parameters, while the Render command and its dialog is to begin renderings. No save function is provided in the Render dialog.*

Deleting preset rendering options

To delete an entry from the custom rendering options list, hold down the Option key while selecting it by name. Upon releasing the mouse button, the system “beep” will sound to confirm the deletion of that entry in the sub-menu list. This event is not undo-able, so you should be sure of your actions.

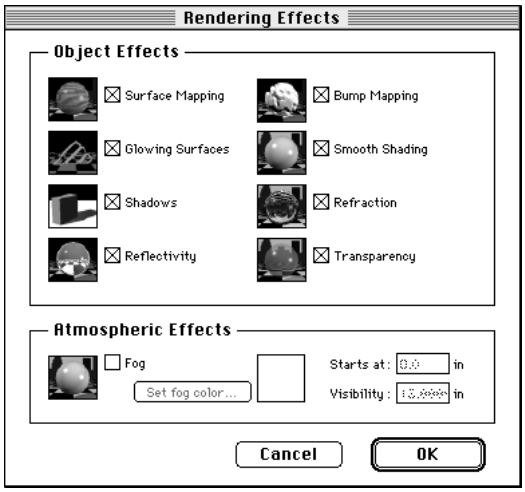
If you do change your mind, you can always add the preset again by manually enabling the same rendering options and saving them to the list.

Rendering options explained

The effects included in a rendered image are determined by the options enabled in the Rendering Options and Rendering Effects dialog. The rendering algorithm used must support the effects chosen. Conversely, if an effect is not enabled in this dialog it will not appear in the rendered image, even if the renderer supports it.

Turning rendering effects on and off

The Rendering Effects dialog contains check boxes to enable or disable the primary effects available in Vision 3d. This dialog is opened by choosing the Rendering Effects command, or by double-clicking on the Camera tool.



Whether or not a specific effect appears in a rendered image also depends on the rendering method chosen. Obviously, you cannot expect to see refraction with anything less than raytracing. Or smooth shading when using the flat shading renderer. On the other hand, if refraction is turned off in this dialog, even if you do use the raytracer, it won't appear in a rendering. The two must be coordinated.

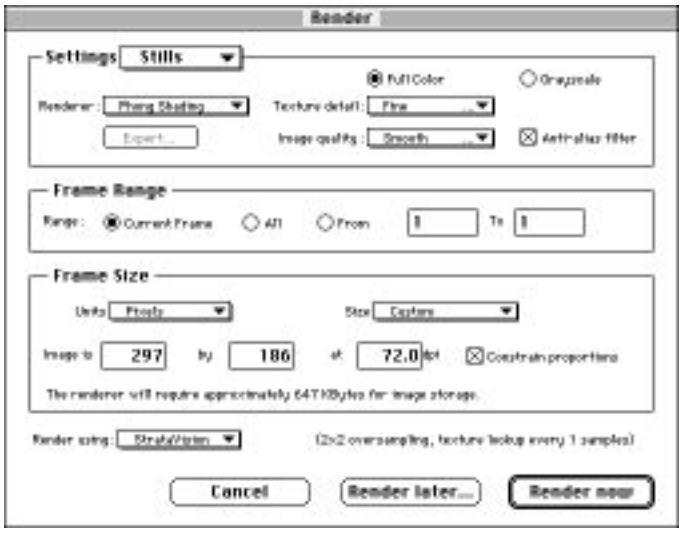
To avoid confusion, each time Vision 3d is launched for a new session, all of the rendering effects, except Fog, are automatically enabled. The on / off status of each effect can be set for the current session only.

Setting rendering parameters

Choosing the Render command initiates a rendering. It is the same as clicking the Camera tool in the active view while holding down the Shift key. Either method defines a rendering window the same size as the active window in which you clicked.

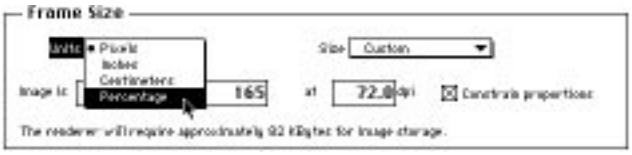
You may not want to render an image the full size of the active view, larger than the active view, or you may wish to crop in tighter, but still have access to the Render dialog to set the rendering parameters before beginning the actual rendering. You can crop the active view by dragging a snapshot marquee with the Camera tool. Holding down the Shift key

as you click-drag the Camera tool allows you to crop and open the Render dialog.



Within the Render dialog are fields to set all the relevant parameters for a rendering. From this dialog, you can adjust the fields listed on the following page.

1. **Frame size.** You may designate pixels, inches or centimeters as the units for the image dimensions. Several pre-defined sizes are provided in the Size field pull-down menu. Selecting a size from the menu automatically resets the image dimensions in the height and width fields. You must return to the Custom setting to be able to edit the image's dimension fields.



Optionally, you may use the Percentage setting in the Units pull-down menu. This allows you to change the size of the rendered image as a percentage of the current size. Thus, when in this mode, entering 100% in either image size field keeps the image's size in that dimension the same. Values larger than 100% enlarge the size of the image; values less than 100% reduce it.

2. **Base resolution.** This field allows you to adjust the number of dots per inch or dots per centimeter in the final image. This setting is the resolution of the final image and the level at which the anti-aliasing option interpolates its super-sampled pixels. Anti-aliasing is enabled with a check box at the top of this dialog.
3. **Render a single frame or multiple frames for an animation.** Each time this dialog is opened the Range radio buttons are active. The current frame is identified by the position of the time pointer on the time line in the Animation palette. If the Length pointer on the Animation palette has never been advanced, there is only one frame available to be rendered, at time zero. If you have laid down event markers for an animation, you can choose to render some, or all, of the frames in the animation based on the animation parameters set on the Animation palette. Two parameters on the Animation palette affect the number of frames that can be chosen for rendering in the Render dialog: The frames per second and the animation length.
4. **Rendering method and quality level.** A pull-down menu provided in this dialog box allows you to choose the rendering method. Other fields in the Settings section of the dialog set the texture(s) sampling rate and use of anti-aliasing.
5. **Render now or later.** You can choose whether to proceed with the rendering now or save a suspended file for later rendering. If you choose to render now, the Render dialog box closes and a rendering window immediately opens to begin the rendering process, as specified by the parameters in the Render dialog box. If you choose to render later, the Save As dialog box appears to name and save a suspended rendering.

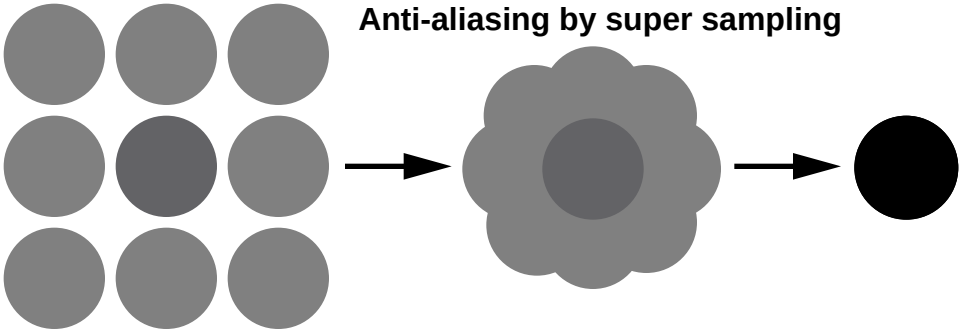
About image resolution and anti-aliasing

A computer-generated image is only as good as the "apparent" smoothness of its edges. Jagged edges usually result from not having enough information (resolution) to accurately describe the image. The process of smoothing edges in an image is called anti-aliasing. In addition to simply rendering at a higher resolution (a.k.a, the "brute

force method"), there are two other ways to achieve a smooth-edged image:

- **Super Sampling.** With this method, the image is rendered with excess resolution. The extra pixels are then interpolated down to the designated base resolution. Super sampling requires that the image resolution be greater than the resolution of the medium on which it is to be eventually displayed or printed.

Anti-aliasing by super sampling



An anti-Aliasing Factor of three generates a three-by-three pixel grid centered around each pixel in the final image.

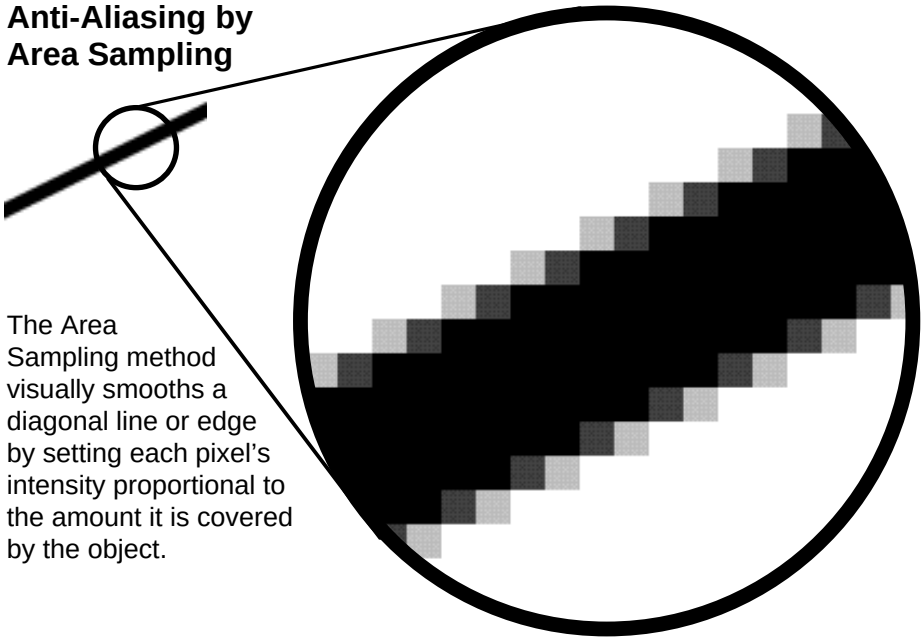
When the rendering is completed, the pixels are interpolated down to the one pixel used in the image.

The final image appears smooth because its pixels are the result of many extra samples taken during rendering.

For example, a 144 dot per inch (d.p.i.) image displayed by QuickDraw on a 72 d.p.i. screen will appear smooth because of the excess data available to the screen. You only see the 72 d.p.i. resolution, but the image appears smooth because of the super sampling effect produced by QuickDraw.
In Vision 3d, this is the same effect of rendering extra pixels and interpolating them down to the finished image.

- **Pixel Averaging.** This method alters the color of the pixels along diagonal lines or edges. The adjustment is proportionally weighted, based on how much of each pixel the line covers.

Anti-Aliasing by Area Sampling



The Area Sampling method visually smooths a diagonal line or edge by setting each pixel's intensity proportional to the amount it is covered by the object.

Pixel averaging uses only the existing pixels at the base resolution to perform its smoothing. Because of that, this method of smoothing images can produce subtle blurring that may cause loss of fine details.

Each method has its advantages and disadvantages. In Vision 3d, both methods are available, although Vision 3d uses super sampling as its primary method of anti-aliasing.

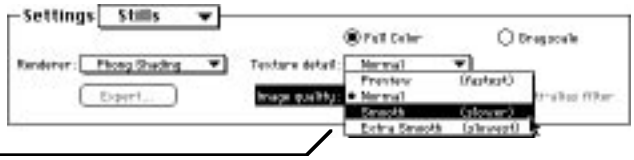
Using the super sampling method

Because super sampling works its smoothing magic through extra resolution, it produces an image with superior sharpness along with the smoothness.

Super sampling anti-aliasing is used when the Image Quality field in the Render dialog is set to "Smooth" or "Extra Smooth." The smooth setting generates two pixels vertically and two pixels horizontally for

each pixel in the final image. The extra smooth setting renders three pixels both directions.

Choosing "Smooth" or "Extra Smooth in the Image Quality menu turns on anti-aliasing



Thus, the intermediate image is four times larger for the smooth setting, and nine times larger for extra smooth. The excess data is then interpolated down to the final image specified by the Frame size and d.p.i. settings in the same Render dialog.

If you render a 72 d.p.i. image with the Smooth image quality setting, the intermediate image is effectively 144 d.p.i., but it is re-sampled down to 72 d.p.i. for the final image. If you asked for a 150 d.p.i. image, with the smooth level of anti-aliasing the intermediate image would be 300 d.p.i. With the extra smooth level of anti-aliasing, the intermediate resolution would be 450 d.p.i.

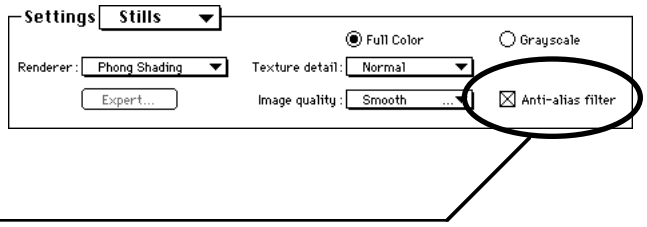
If the anti-aliasing factor is set to extra smooth, an alert is displayed warning of the longer rendering times required for the extra resolution. This same alert can appear in other places throughout the program, whenever a setting is made that is considered atypical. It is only intended to make you aware of the significance of the choices you make.

Using the pixel averaging method

This method of anti-aliasing can produce visually pleasing images without the need for higher resolution and larger file sizes. Pixel averaging is especially useful in situations where extra resolution via super sampled pixels are not available, i.e. — existing images (rendered or otherwise) at low resolution.

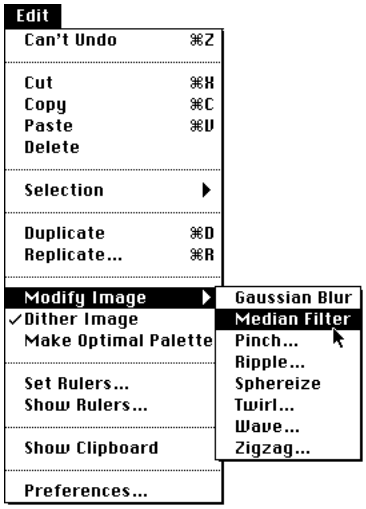
During rendering in Vision 3d, pixel averaging anti-aliasing can be added to the super sampling to obtain a vastly superior image. This is enabled with the checkbox to the right of the Image Quality menu.

A pixel averaging anti-alias filter becomes available when the Image Quality field is set to Smooth or Extra Smooth



Note that this feature is only available when the Image Quality pull-down menu is set to Smooth or Extra Smooth. This is an image quality consideration. If Vision's anti-aliasing filter is used alone — on a simple, non-anti-aliased image — the image will become too blurry. But when used in conjunction with the super sampling, the resulting image is incredibly good.

Pixel averaging anti-aliasing can also be supported in Vision 3d as a post-rendering filter or extension. In that manner, images from any source can be opened in the program and anti-aliased using this method. Vision 3d includes two built-in filters for this purpose: 1) Gaussian Blur and 2) Median Filter. Other anti-aliasing filters may become commercially available from Strata Incorporated or third-party sources.



Additionally, you may use one or more “specialty” filters to further modify images for various purposes. These are also accessed through the Modify Image command in the Edit menu. All filters work on the entire image. There is no provision to select limited areas of an image in Vision 3d. That capability is available in other applications.

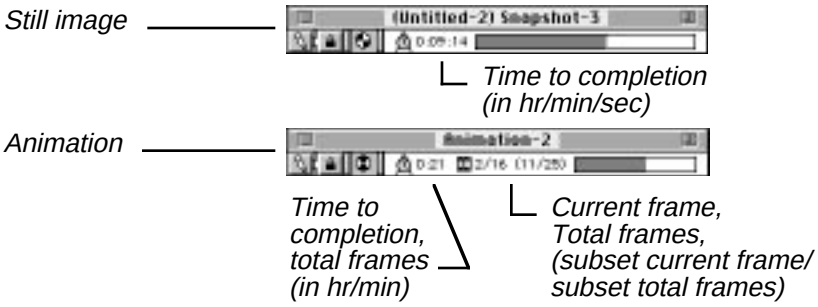
Specialty filters may also be obtained as extensions. For example, Vision 3d can accept third-party “plug-ins” marketed for use with Photoshop. To install, copy them into the Extensions folder residing in the StrataVision 3d application folder. Under System 7, you can put aliases of the plug-ins in the Extensions folder and leave the originals in the Photoshop folder, thereby avoiding using excessive hard disk space to store duplicate files in two places.

Extension image filters appear in the Modify Image sub-menu, when present.

Rendering windows

Rendering windows are discussed in chapters 1 and 7. They are mentioned here in the interest of clarifying the information provided at the top of a rendering window while it is in the process of rendering an image.

Feedback found in rendering windows:



When a rendering is begun, a sequence of events occurs internally to allocate memory for the rendering. One of the events is to open a rendering window from which you can monitor the progress of the rendering. Several pieces of information are provided in the feedback

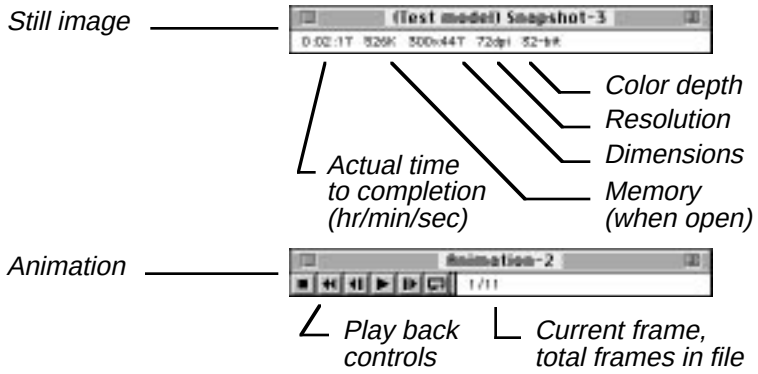
area of a rendering window. That information varies slightly, depending on whether the rendering is a still image or part of an animation sequence. The differences are itemized in the previous illustration.

Image Display Windows

Image windows are also discussed in chapter 1 and 7. They are mentioned here again to draw your attention to the information and/or controls provided at the top of the window. The contents of this portion of the window differs with the type of file, as shown in the illustration below.

When a rendering is completed, the rendering window automatically converts to one of the image windows shown here. If the rendering was begun in the current session from a suspended rendering, the image is saved automatically. Otherwise, Vision 3d waits for you to manually save it. In either case, the image window remains open until you close it.

Feedback found in image windows:



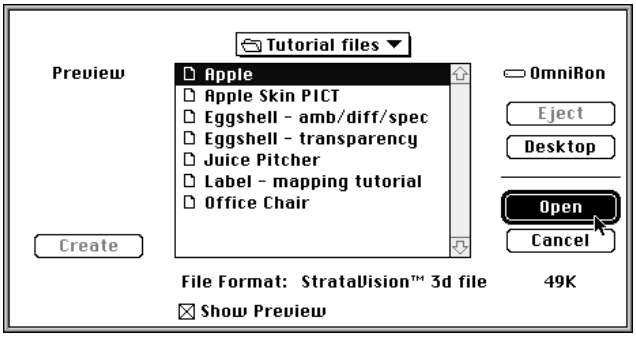
Of course, one of the image windows is also used when you open an image or animation that has been saved previously. Images and animations from any source can be opened, not just files saved from Vision 3d, as long as they are in a format recognized by the software.

Quick tutorial

The purpose of this quick tutorial is to acquaint you with some of the rendering methods in Vision 3d. You will use a model, titled “Apple,” to render the same object using different rendering methods in order to compare and contrast the results. You will examine the different renderings in terms of appearance, image detail, and length of time required to complete the renderings.

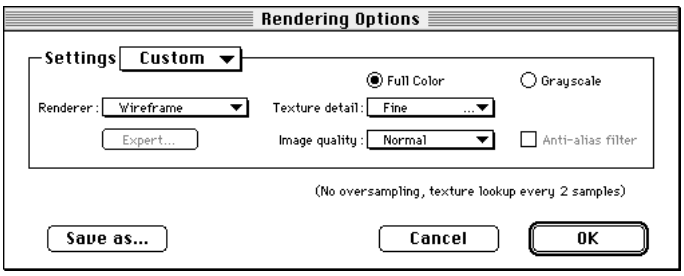
The model is that of a shiny red apple. Notice that this apple is not just a simple symmetrically-lathed object, but a realistic asymmetrical construction of an apple. The apple actually started out as a sphere primitive that was “molded” in 3d Sculpter for a more apple-like appearance. The leaves were also made with the 3d Sculpter modeler, and the apple surface is made from a solid texture. Despite all of the explicit geometry, the modeling is actually quite efficient.

1. **In the StrataVision 3d Tutorial folder, find the file titled “Apple” and open it.** Pull down the File menu and choose the “Open” command. Then navigate to the Tutorial folder. Find and open the Apple file.

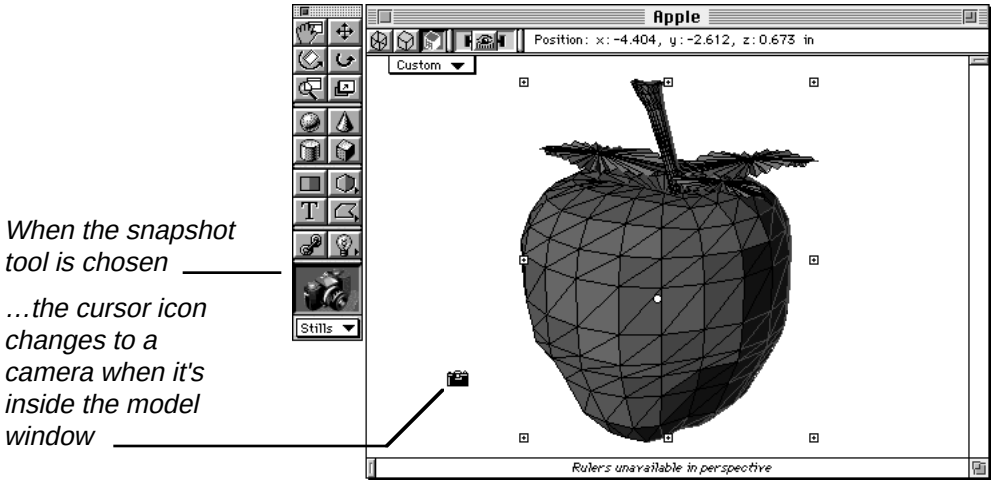


When the file opens, you should be in a custom view with the three-position Lens control set on Normal and the display method set to wireframe. Use the "Fit Views to All" command if necessary to show all of the apple in the view. The custom view used here is 18° up from the Front view to enhance the perspective of the apple when it is rendered.

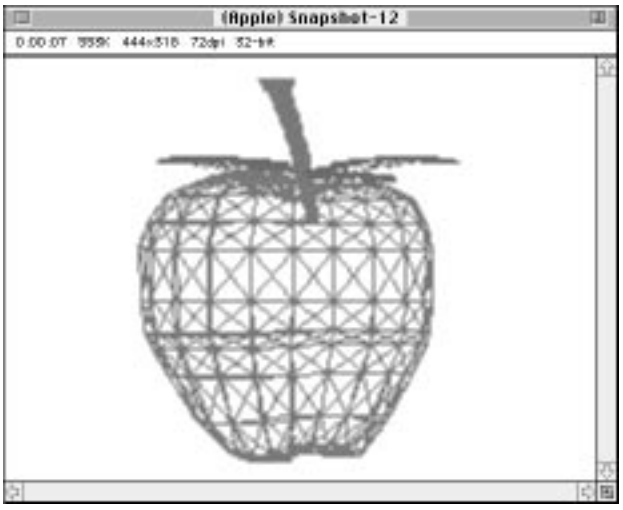
2. **Pull down the Rendering menu and choose the Rendering Options command.** Vision 3d opens the Rendering Options dialog.
3. **In the Rendering Options dialog, set the Renderer pull-down menu to Wireframe.** Make sure the Image Quality pull-down menu is set to Normal. Then click on the OK button. For now, you will ignore the Texture Detail field because Wireframe rendering does not support textures.



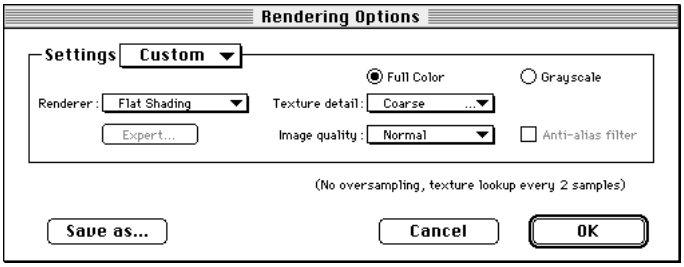
4. **Select the large camera snapshot tool at the bottom of the tool palette.** Notice how the icon changes to a little camera when you move the cursor into the model window when this tool selected.


When the snapshot tool is chosen
…the cursor icon changes to a camera when it's inside the model window
Apple
Position: x: -4.404, y: -2.612, z: 0.673 in
Custom
Stills
Rulers unavailable in perspective

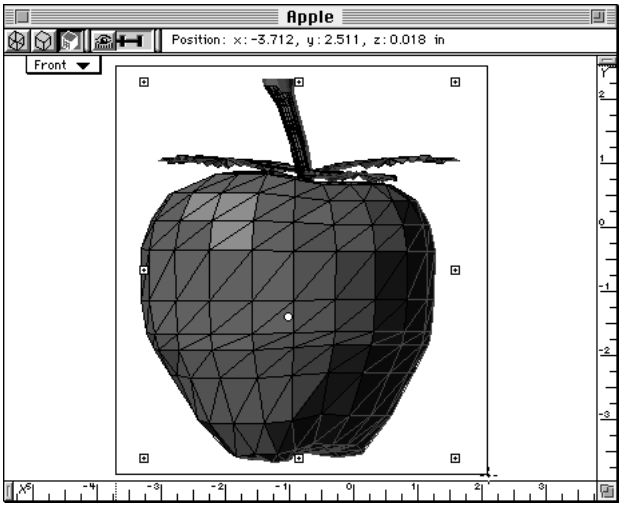
5. **Click once in the model window.** Vision 3d begins to render the apple. Since this is a very simple rendering, it will finish quickly. The finished rendering should look similar to the following illustration.



6. **Save the rendering.** Name it “Apple-Wireframe.” Choose the PICT file format. You’re saving this rendering so that when you finish this exercise, you can open all your renderings of this model and compare them side-by-side to see the differences between the various rendering methods.
   Once the image is saved, close the image window.
7. **With the model window active again, open the Rendering Options dialog again.** This time choose Flat Shading from the Renderer pull-down menu. Image Quality should be on Normal. Once again, you can ignore Texture Detail settings. Click on the OK button.



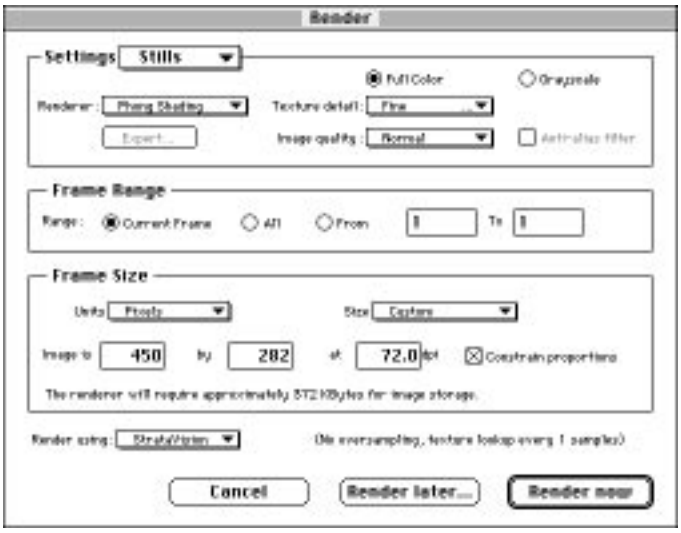
8. **Choose the snapshot tool again, then in the model window click-and-drag a marquee around the apple as illustrated.** When you release the mouse button, Vision 3d begins to render the apple using the Flat Shading rendering method.



The click-drag method of initiating a rendering allows you to render only that portion of the model window you wish to have rendered. By contrast, the single click method renders the entire model window.

Notice that the Flat Shading rendering method provides more detail than the Wireframe method. The apple has a surface in this rendering, not just a frame outline as with the wireframe rendering method. You will also notice it takes slightly longer for a Flat Shading rendering to finish than for a wireframe rendering, although the time is still very short. That's the rendering trade-off: the more detail in a rendering, the more calculations required to create the image, and the longer it takes for Vision 3d to complete the rendering task.

9. **When the rendering finishes, save it to your hard drive**. Call the file "Apple-Flat Shaded."
10. **Open the Rendering Options dialog again.** Set the Renderer to Gouraud Shading. (Gouraud is named after the man who invented this method of rendering 3-D objects.) Make sure the Image Quality is on Normal. Also, this time set the Texture detail field to Normal. Then click on the OK button.
11. **Render the apple using the Gouraud method.** Use either the single click or the click-drag method to start the rendering. When the rendering finishes, again notice that the Gouraud rendering method provides even more detail than the previous rendering methods. The apple has a smooth surface in this rendering, instead of the faceted surface you saw in the Flat Shading rendering. This rendering also took slightly longer to complete than the Flat Shaded rendering. Save the rendering to your hard drive. Name it "Apple-Gouraud."
12. **Time for a change: Hold down the Shift key while you click once in the window.** The Render dialog appears. Notice how the top part of this dialog is very similar to the Render Options dialog.
    This time, set the Renderer to Phong Shading. (Like Gouraud, Phong is named after the man who invented *this* method of rendering.) Image Quality should still be Normal, but now set the Texture Detail field on Fine.



Because the Phong renderer is the lowest rendering level that supports textures, you now can exercise control over the rendered quality of textures independent of the renderer itself. As the different texture detail settings explain, the less texture detail you render, the faster the rendering will finish. The same is true of the Image Quality settings: The higher the image quality setting, the longer the rendering will take to complete.

13. **Click on the Render Now button.** When the rendering finishes, again the rendering trade-off is quite noticeable. The completed rendering looks more realistic than ever, but took longer to finish. Save the finished rendering to your hard drive. Call it “Apple-Phong.”
14. **Time for another change: Click on the pull-down menu at the bottom of the Tool palette.** Choose Effects, then select Watercolor from the sub-menu.
15. **Render the apple using the RayPainting rendering method.** This rendering method is a very interesting variation from the other rendering methods. This rendering algorithm is designed to simulate artistic effects rather than photo realism. Save the finished rendering to your hard drive; call it “Apple-RayPaint 1.”

16. **Repeat steps 14 and 15.** This time, choose Pencil on Rag from the Effects sub-menu. This produces a completely different look than the watercolor pre-set. Save this image as "Apple-RayPaint 2."
17. **Click on the Pull-down menu on the Tool palette.** Choose Stills, then select Better from the sub-menu.
18. **Render the apple using the Raytracing rendering method.** This rendering method takes longest to complete of all you will use in this exercise, and as you now know, it's also the most realistic. Save the finished rendering to your hard drive; call it "Apple-Raytracing."

Now open all the saved renderings using Vision 3d or another application and compare them. Each rendering method you tried in this exercise has provided progressively more detail, and thus a better, more real-looking image than the previous method. RayPainting is an interesting and useful variation from the other rendering methods with realism as their priority.

The time trade-off was also clear. The wireframe rendering finished fastest. The raytracing took the longest. However, raytracing is also the most real-looking of the rendering methods. From this you've learned about the rendering trade-off: The more realistic the image, the longer it takes to render.

You can see it's wise to render an image only with as much detail and realism as needed to accomplish the project. It's not necessary to raytrace every image — using high quality and fine detail settings for everything. In many cases, the other rendering methods will produce a sufficiently realistic image while finishing the rendering task much more quickly.

Choosing the appropriate rendering method

The rendering process involves giving objects shading. A finished image is made of pixels, each with a calculated color derived from the interaction of object surfaces with light sources inserted in the model. Each pixel in the rendered image requires four bytes (32 bits) to define if the image is in color, or one byte (8 bits) if only gray scale is used. Note that there is no speed improvement by rendering in grayscale, but it will require less memory during rendering, as well as less disk space after it is saved.

Once rendered, an image may be saved in various color depths and formats for further use in other applications. It may also be output directly from Vision 3d using the Print command to obtain a hard copy.

Vision 3d uses the System's Print Manager tools to print to any Chooser-selectable device. As you may know, that device's print driver must be installed in the System to choose it for printing. The driver provides default settings to Vision 3d for output. Other applications may provide better output control.

Vision 3d contains ten different rendering algorithms in its repertoire to provide images of a model with different degrees of realism, and for various purposes. The extended choice allows you to balance the image quality requirements against the time required to generate it. If a lesser renderer can accomplish the job in acceptable quality, why waste time on an image that takes longer to render?

The renderers provided in Vision 3d are: 1) wireframe, 2) solid (wireframe with hidden lines removed), 3) quick shaded, 4) flat shaded, 5) Gouraud, 6) Phong Shading, 7) RayPainting, 8) Raytracing, 9) Phong with Shadows, and 10) RenderScript. While the lower-level renderers — wireframe, solid, and quick shaded — are used primarily for displaying objects in the modeling window and camera window views, they are also considered renderers in their own right. Images in the modeling windows created by the wireframe, solid or quick shaded renderers may be saved as QuickDraw PICT files containing draw-type vector data for the image. The higher-level algorithms use surface normals to calculate the smooth shading of surfaces and produce a pixel-based image in the resolution specified.

As described elsewhere in this manual, a "normal" is a vector (or set of vectors) attached to each polygon which indicates the orientation of the surface. Knowing which way a polygon is facing is very important for the renderers. The higher-level renderers use these normal vectors from either the polygon or the vertices of the polygon to produce smooth-shaded surfaces essential for added imaging realism.

How raytracing works

More accurately termed "backward ray tracing," the raytracing algorithm operates by following each ray of light striking a rendering's view plane (defined by the active window) backward through the model to their source(s). If a ray comes from light reflecting off an object's surface, the texture applied to that object must be consulted to determine what that reflected light should look like. That may include surface maps, reflectivity, shininess, or many other lighting factors. If the ray of light is coming from another object that is reflecting in the first object's

surface, then the ray must be pursued further backward from this surface to determine whether it came from another surface or from a light source.

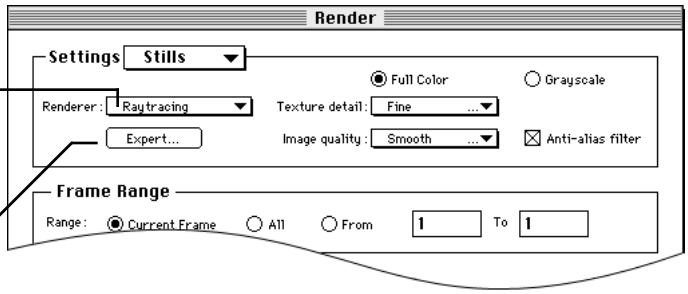
This backward tracing continues until each light ray is accounted for; either ending at a light source or passing out of the scene. The minimum number of rays that must be traced is equal to the number of pixels in the image being rendered, plus any rays spawned to handle reflective surfaces, transparent surfaces and shadows. Thus, the larger the image and the higher its resolution, the longer the rendering time. A 288 d.p.i. image, for example, has four times as many rays to trace as a 72 d.p.i. image of the same dimensions.

CHANGING RAYTRACING PARAMETERS

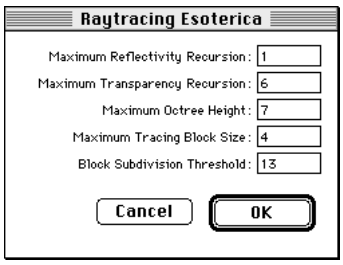
When you choose to use the Raytracer (by choosing it from the "Renderer" pull-down menu), the "Expert" button becomes active just below the Renderer field. This button allows you to open the Raytracing Esoterica dialog box. Operating settings for the Raytracer are displayed in this dialog box.

Choosing the Raytracing renderer

…allows access to the "Raytracing Esoterica" dialog through this button



The default values (that are present in this dialog box when Vision 3d is first installed) are settings designed by engineers at Strata Inc. to deliver the most efficient, best all-around performance from the raytracing rendering algorithm. The fields have been made accessible to allow for customizing the raytracer as needed for special imaging requirements. Before changing these settings, however, you should thoroughly understand how these fields control the algorithm. Otherwise, you may adversely affect the speed and efficiency of the raytracer.



If you do change the settings in this dialog, then want to return to the factory settings, just choose one of the pre-set rendering options from the Settings pull-down menu in the Render dialog. The values in this dialog will revert to those shipped with the software.

INTELLIGENT RENDERING

To make the raytracing process more efficient, Vision 3d assigns priorities in the method it goes about searching for the sources of light rays. This prevents or limits chasing after rays of light that do not materially contribute to the image.

One method used is to divide the model space into smaller volume elements, or “voxels.” The model space is first sub-divided along the three axes, defining eight smaller cubes of the space. Each of those cubes may then be subdivided into another eight smaller cubes, depending on whether there are objects found in them. No objects, no further sub-division and no more time spent there.

Not all cubes are subdivided the same amount. The actual number of times one of the original eight cubes is subdivided ultimately depends on the number of objects in it, but is also controlled by a value in the Raytracing Esoterica dialog box (explained later). The various levels of subdivision is called an octree structure. “Octal” means eight, which is the number of smaller cubes created each time a larger cube is subdivided on each axis, and a schematic representation of the subdivision process takes on a tree-like appearance.

The subdivision may be repeated as often as needed to define small enough volume elements (with few enough objects in each voxel) that the Raytracer can interrogate them quickly and efficiently. By comparing the location of objects in the model to the voxels in each octree level, the

algorithm can know where to spend time tracing rays of light and which areas it can ignore.

FIELD DESCRIPTIONS

The following descriptions of the fields in the Raytracing Esoterica dialog is not intended as a complete explanation of how Vision's raytracing algorithm works. Entire textbooks have been written on the concept of raytracing. Some of these are listed at the back of this manual in the Bibliography of the Appendices section. Of course, Vision's raytracer is proprietary, so it has been enhanced and may operate just a little differently than the public domain versions.

This discussion is meant only as a guide to help develop your own judgment in adjusting the raytracing parameters for special rendering situations. You should realize that the optimal values for the raytracer can vary from project to project, depending on model size, available memory, the type of surface maps in the textures used, etc. There are no wrong settings, just degrees of rightness. Obviously, anything that slows down rendering speed without significantly improving image quality is less "right" than a configuration that gains performance while maintaining the excellent image quality for which Vision 3d is famous. Arriving at the optimal raytracing values is often a matter of compromise.

Of course, you are encouraged to experiment with the numbers in this dialog. Do so with the intent of obtaining faster rendering or better quality (or both), once you understand what to look for. Understand that these two goals are usually mutually exclusive. You should develop a methodical approach to any changes you make and do some benchmark tests to compare relative performance.

The fields in the Raytracing Esoterica dialog and their effect on rendering are:

- **Maximum Reflectivity Recursion.** This field determines how far a ray of light is traced as it reflects from surface to surface. In visual terms, it determines how many times an object will be reflected in the surfaces of other objects. The default setting is one. For most models this is adequate. Increasing this setting usually adds to the rendering time. But if you are modeling a "House of Mirrors," you will want to set this value higher.
- **Maximum Transparency Recursion.** This field's setting only affects models with multiple transparent objects positioned in front of each other. It determines how many layers of transparent surfaces the

Raytracer will look through to see if there is something behind before giving up. If more transparency layers exist than this setting allows the raytracer to look through, that area of the image will appear black. The default setting is six.

- **Maximum Octree Height:.** As discussed earlier, Vision's Raytracer sets up several octree levels during the rendering operation. An Octree is an internal data structure used to subdivide the model space for more efficient handling and to simplify the ray tracing process. Initially, all objects are contained in one large cube or model space. The model space is adaptively subdivided by a process where each cube on one level yields eight smaller cubes on the next level. The actual number of octree levels created for a particular rendering depends on the complexity of the model.
  Theoretically, the higher the number, the faster the raytracer will run. There is a point, however, when the octree structure can become too big and it begins to take more and more time to maintain it; negating any gains in sub-division efficiency.
  This field determines the maximum number of levels in the octree to be defined for a rendering operation. The higher this value, the larger the memory partition for Vision 3d needs to be. The octree structure must be set up within the application's memory partition. Higher numbers divide the model space more finely, but take longer to set up and sort through. A lower value uses less memory, but may decrease the Raytracer's efficiency (meaning less speed).
- **Maximum Tracing Block Size.** In the creation of an image, the renderer starts with large blocks of pixels which are refined smaller and smaller during the rendering process. The refinement continues on every pixel block until the maximum block size is reached. At that point, the Block Subdivision Threshold setting decides which blocks to refine smaller. A setting of four in this field means a four-pixel by four-pixel block.
  Larger block sizes render much faster, but may sometimes cause the raytracer to miss fine lines. If your model has lots of detail work — either as surface maps or as explicit object geometry — it is often desirable to set this field smaller. In models that do not have fine details, a larger block size is usually preferred.
  Setting the maximum block size smaller will not only insure picking up the fine details in an image, but may actually render faster in those cases where it is needed. That's because starting with smaller block sizes at the start of a rendering doesn't waste time later having to go back to find the detail that was missed, if it is found at all.

A tracing block size that is too large for the situation usually shows up disappearing holes in 3-D text or the one-pixel lines in surface maps not showing up consistently. This becomes especially obvious in animations, where the detail is present in one frame, but not the next.

- **Block Subdivision Threshold (percent)**: When the maximum tracing block size is reached, the raytracer scans the image block-by-block to see if it needs to be refined further. It usually does. If two adjacent blocks of pixels have a difference in color values, this field decides the variation threshold — when it is significant enough for the raytracer to pause and further refine the two blocks to resolve the difference. It will continue this process until all adjacent blocks (pixels) are within the allowed percent of variation.
  Setting the value higher increases rendering speed, but may cause noticeable color banding of certain colors on curved, shaded surfaces. Setting the value lower forces all block sizes to be resolved finer, even if banding is not visible.

Raytracing is the glamour algorithm in the Vision 3d line up. There is a tendency to want to use raytracing for all rendering because it looks so good. But many of those extra effects are also available with the Phong algorithm. In many scenes set up for Phong rendering with an environment map, the result cannot be distinguished from Raytracing.

Effects like object reflectivity and refraction come at a computational cost in raytracing. Not every image needs to be raytraced; not every image is worth the investment in time. If tempted to jump right to the most realistic rendering effects, you should stop and analyze the actual requirements of the image. Refraction and shadows are the only effects not supported in some manner in one of the other (faster) algorithms. If refraction and/or shadows will not make a major contribution to the finished image, perhaps Phong or one of the other renderers would do the job as well, and a lot quicker.

How RayPainting™ works

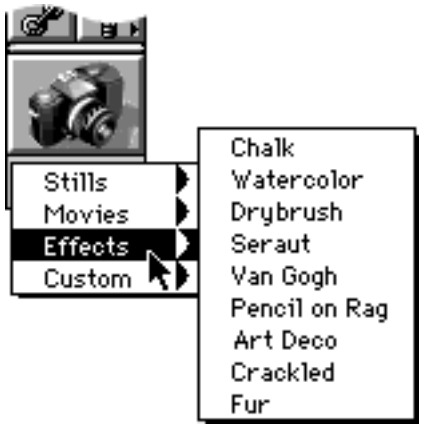
RayPainting is a diversion from the stereotypical 3-D rendering algorithm. While it uses the same basic rendering method as the raytracer, it also adds some very special and dynamic artistic effects that put some creativity back into the rendering process. One thing you will notice right away is how much faster this rendering method is than raytracing. It is often faster than the Phong renderer. This is especially true with higher resolution renderings.

More than any other renderer, the RayPainter begs to be explored. You'll be amazed at how many different effects you can come up with as you try variations of the different settings that control the RayPainter. As you do, you will notice that this renderer is more than just a fancy image filter. It is a true 3-D brush whose strokes follow the geometry and contours of objects in ways that no image filter can begin to emulate.

You do not set the RayPainting rendering parameters by the numbers, as you would with the raytracer. RayPainting is concerned more with artistic effects than realistic detail. RayPainting rendering settings are made by choosing items like canvas type, brush type, stroke gap, "jitter," etc.

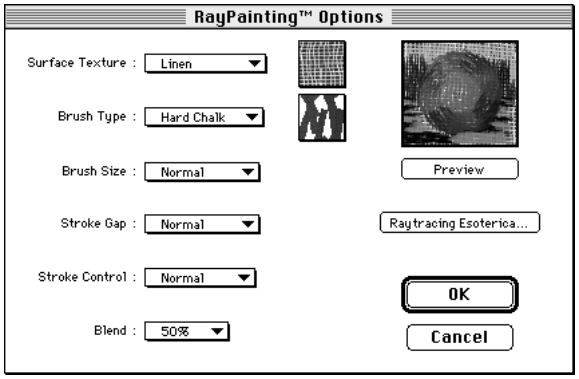
To use the RayPainter, you can begin by trying the several pre-sets available through the pull-down menu on the Tool Palette.

The entire Effects sub-menu uses the RayPainting renderer to bring artistic creativity to 3-D rendering



To go beyond those effects built-in to Vision 3d, just click on the Tool palette's pull-down menu, the choose the "Custom" entry in the Custom sub-menu (Custom • Custom) …or choose the Rendering Options command in the Rendering menu. Either method opens the Rendering Options dialog box.

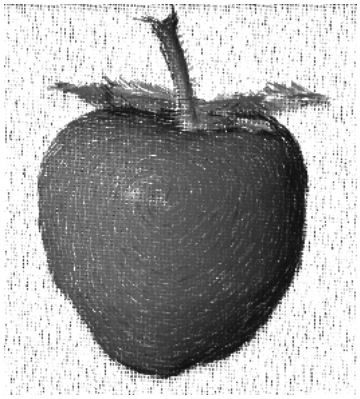
Choosing RayPainting from the Renderer pull-down menu, makes the Options button just below this field active. Clicking on the Options button brings up the RayPainting Options dialog.



This is where you can experiment with the RayPainting parameters and try them on the preview model right there in the options dialog. The preview is a special model built-in the RayPainting code to show the various effects you choose to best advantage. While you are learning the impact each field has on the RayPainting effect, it is best to develop a methodical approach. Change one thing at a time and see what effect it has on the preview image.

Image rendered with RayPainting

The canvas was set to linen; the brush type to hard chalk. All other field were left at normal. The blend was set to 50%.



When you find a combination of RayPainting settings you want to keep, you can click on the OK button to exit this dialog and save it by

name in the Rendering Options dialog, using the Save As button. A collection of rendering option settings saved in this manner appear in the rendering options pull-down menu under the “Custom” sub-menu.

RayPainting Esoterica

The RayPainting Options dialog also contains a button titled Raytracing Esoterica. Clicking on this button displays the same Esoterica dialog you have seen in the discussion on Raytracing. But notice that the numbers in the fields are different. They are definitely “looser” in appearance This is consistent with the type of images the RayPainter produces. As you know, the RayPainting renderer uses the raytracing algorithms to do many of its calculations. Therefore, the raytracing settings in this dialog also apply to the RayPainter.

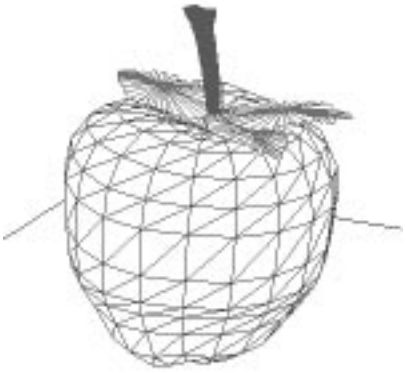
NOTE: *The numbers in the Raytracing Esoterica dialog are changed to RayPainting specifications when you select one of the Effects pre-sets from the pull-down menu on the Tool palette. If you then plan to use the Raytracer again, IT IS IMPORTANT that you set the Esoterica fields back to those used for raytracing. The easiest way to do this is to choose “Stills•Better” or “Stills•Best” from the same pull-down menu. If you fail to do this, the raytracings you may do later will be noticeably less photo-realistic than you might expect. They may even have visual flaws in them.*

As mentioned at the first of this discussion about RayPainting, the numbers in the Esoterica dialog are not as important as the settings in the RayPainting Options dialog. These are the controls that give RayPainting its unique look.

Comparison of the rendering algorithms

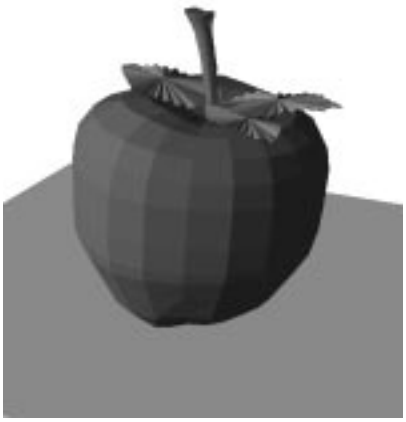
The following rendered examples and comparisons of features is provided to help make better choices of the most appropriate rendering method. A chart is also included after the individual discussions of each renderer.

- **Wireframe.** This is the simplest rendering method used by Vision 3d. It is primarily a display method for the modeling views, but may also be used for rendering separate images. It generates a line representation of object shapes in a model for displaying on the 2-D views of the screen or outputting to any QuickDraw-compatible printer. No surfaces, lighting, or surface shading are calculated or implied. In wireframe, surface-bearing polygons and hollow polygons are displayed the same way.



- **Hidden Line Removed.** This display method removes the hidden lines from wire frame representations before displaying them in the views or rendering in an image. Wire objects presented this way

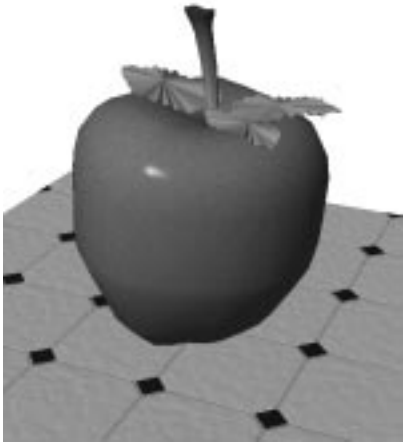
appear solid, with implied surfaces, but without illumination of the surface. Solid displays make it much easier to discern the orientation of objects.



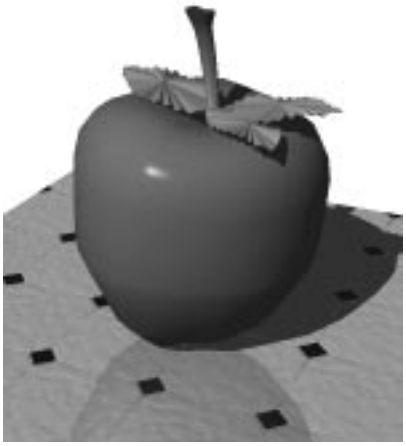
- **Quick Shading.** This display method calculates the actual surface orientation for each polygon. It also shows the illumination intensity for those polygons, providing rudimentary shading of the object(s). Shaded images require the most time to redraw in modeling views whenever changes are made in the model. This method is the highest quality image available in the modeling views. Like the previous two display methods, it can also be used to render images and / or provide QuickDraw-compatible draw-type PICTs that can be saved or printed.

- **Flat Shading.** This rendering algorithm is sometimes called constant shading. It calculates a single color for each face on the surface. The orientation of a surface relative to the light source(s) illuminating it are also factored into the color as an intensity value.
  Flat shading is relatively fast. But, because it uses only one lighting calculation per polygon, the resulting rendering is obviously faceted in appearance. This renderer is most often used for preview-type images and animation development.

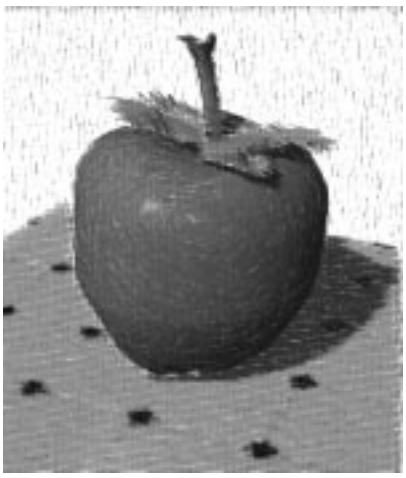
- **Gouraud Shading.** Gouraud is an intensity interpolation renderer. Its algorithm calculates the shading of a polygon by interpolating color intensity from the vertices of shared polygons (in triangulated mesh form), then averaging the shading across the surface of each polygon. Rather than using a single polygon normal, like the previous renderers, Gouraud shading uses the normals at each vertex. The resulting image is smoothly shaded, but can have artifacts at the polygon boundaries. The artifacts are most noticeable in areas around specular highlights on the object's surface.
  Only surfaces constructed as a triangulated mesh, in which vertices are shared by adjacent polygons, have the requisite vertex normals. If no normals are available at the vertex points, the renderer cannot perform a smooth shading interpolation.



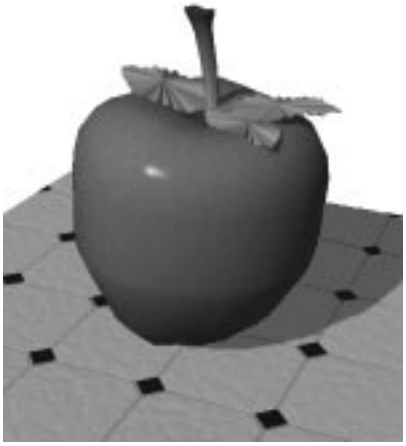
- **Phong Shading.** The Phong rendering algorithm is named for the person who developed this shading technique, Phong Bui Tuong. Phong is a normal vector interpolation shader. It evaluates the position of the vertices, with their associated normals, and interpolates a complete new set of normals across the entire surface of the object. These normals are then used to calculate an intensity for each pixel in the rendered image of that object. The ability to calculate each surface pixel's color also allows Phong to include surface mapping and environment mapping in its rendering effects. Phong shading takes longer than the lower level algorithms, but produces a much better result. However, Vision's version of the Phong renderer is somewhat faster than traditional implementations because it uses a two-pass approach to the rendering operation. The first pass determines what surfaces are visible from the current camera position so the second pass doesn't waste time rendering hidden surfaces (only to cover them up with other surfaces). The Phong renderer has been further enhanced to support environment reflections on objects and more realistic specular highlights from metallic surfaces.



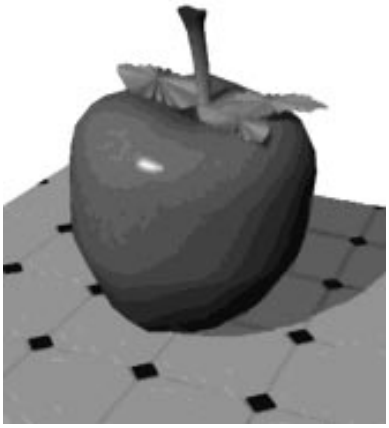
- **Raytracing.** This renderer traces rays of light backward through a model to determine how the image should appear from the chosen viewing position. If the light ray comes from a surface, the renderer then calculates where the light that reflected off that surface originated. In some models, it is possible to trace light rays through multiple reflections until finding the source. Raytracing brings true reflectivity, refractive transparency, and shadows to the imaging calculations. It is more computationally intensive than lower level algorithms, and thus takes longer. The results, however, can be stunning. Vision's ray tracer employs the same shading methods as Phong and supports all the surface mapping types.



- **RayPainting.** This algorithm is unique among the rendering methods available to you in Vision 3d. It is intended as an artistic and special effects renderer, rather than a photo-realistic one like the others. It supports all of the rendering effects, including smooth shading, surface mapping, reflectivity, refraction, shadows, etc. But also allows you to set a surface (linen, rice paper, canvas, etc.), a brush type (hard chalk, crayon, fur, oil, etc.), a brush size, stroke gap, and more. The result is a rendering method that can let you be very creative. The images appear to have been drawn with 3-D brush strokes, which is exactly what it does.



- **Phong with Shadows.** There is a separate renderer selection named "Phong with Shadows." This method supports all the same rendering effects as Phong, plus shadows. Of course, the Shadows checkbox in the Rendering Effects dialog must be checked to get shadows with any of our renderers. Keep in mind that in many situations, rendering with the Raytracer to get shadows may be faster than using Phong with Shadows.



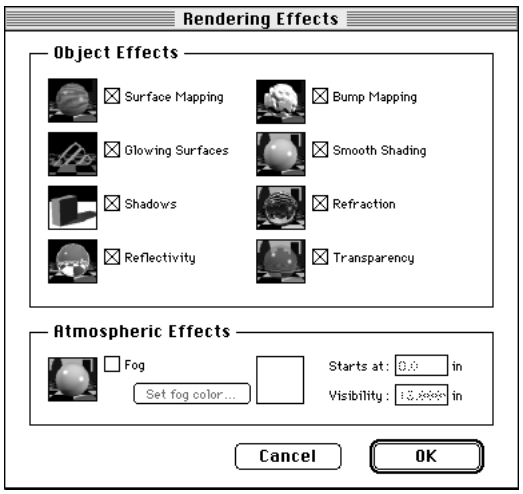
- **RenderScript™.** RenderScript uses a hybrid polygonal renderer that supports Vision's photorealistic rendering effects, such as surface mapping, smooth shading, glowing surfaces, and shadows. This rendering method is based on Phong with Shadows. It is output as an EPS (Encapsulated PostScript™) file with bézier regions. Users who are already familiar with EPS files know that working with PostScript can require a great deal of time, so please be patient. This rendering technique can be used to create surrealistic images which can be scaled to any size, from postage stamp to billboard.

All the renderers are integral with the modeling and lighting tools; making the rendering process a fully interactive tool of equal usefulness to the modeling process. In fact, rendering preview images of specific areas of a scene to adjust maps and lighting can be a very important aspect in the overall scene assembly.

***Note**: RenderScript™ is a new addition to Vision 3d. For more detailed information on this feature, please see the ReadMe file on Installation Disk 1.*

Charts of rendering effects supported (by rendering method)

The effects to be included in a rendering are selected through the Rendering Effects dialog box. This dialog is opened by choosing the Rendering Effects command in the Rendering menu, or by double-clicking on the Camera tool in the Tool palette. The effects available are typically an on / off type of choice. Some choices, such as Fog, require additional parameters which may be set through fields in this dialog.



Note that at the beginning of each session in Vision 3d, all of the rendering effects are turned on except Fog. You must open the Rendering Effects dialog to turn off the effects you do not want or need to use, which can save rendering time, or to turn on Fog. If you disable one or more rendering effects during a session and quit with them off, they are on again at the beginning of the next session.

You should use only the rendering method and effects that actually contribute to the image. In other words, why use Phong renderer if Gouraud will do just as well; or Raytracing when Phong will produce the same image?

	Wireframe	Solid	Quick Shaded	Flat Shaded	Gourand	Phong	Raytracing	RayPainting™	Phong with Shadows	RenderScript™
Anti-Aliasing	●	●	●	●	●	●	●	●	●	●
Draw (Vector) PICTs	●	●	●							
Fog Effects				●	●	●	●	●	●	●
Glowing Surfaces			●	●	●	●	●	●	●	●
Pixel (Raster) PICTs	●	●	●	●	●	●	●	●	●	●
Reflectivity						●	●	●	●	●
Refraction							●	●		
Shadows							●	●	●	●
Smooth Shading					●	●	●	●	●	●
Surface Mapping						●	●	●	●	●
Transparency					●	●	●	●	●	●
3-D Brushes								●		

Enabling a specific rendering effect is only half the process. You must also choose a rendering method that supports that effect. This is done through either the Rendering Options or Rendering dialogs, as discussed earlier in this chapter.

Alternately, you can choose one of the pre-set rendering options from the Tool palette's pull down menu. It will automatically configure the

rendering options for you, based on the type of rendering and quality level you request.

Commands used with rendered images

The following commands are used with the rendering and image/animation display features of the software. The list of commands is alphabetical, and an explanation is included with each item in this section.

Keyboard equivalents are shown on those commands that have been given one in the standard configuration. You may change or add a Command key shortcut on any item in any menu or sub-menu through the Customize Menus command. This feature is explained in Chapter 1.

Dither Image command (Edit menu)

The Dither command allows you to choose whether an image being displayed on a monitor or output to a printer is to appear with dithering or without dithering. The Dither command is available whenever a pixel-based color image is contained in the active window. The command is enabled if a check mark appears before the command when the Edit menu is pulled down. To toggle the command on or off, select the command and release the mouse button.

Dithering is a diffusion technique used by 32-Bit Color QuickDraw to allow images of varying color depths to be displayed on monitors of any color depths. The effect is most noticeable when displaying a full-color image (32-Bit, 16.7 million colors) on a monitor with fewer colors available to it, usually 8-bit (256 colors or grays) or monochrome. The dithering process simulates actual colors in the image by mixing white or black patterns with colors available on the monitor.

In Vision 3d, you can choose to display an image without dithering, meaning only the pure colors that are available on the monitor at that time will be used. Sometimes this improves the image quality, but most often there will be obvious color artifacts. This option is usually not recommended.

Environment Effects command (Rendering menu, ⌘ H)

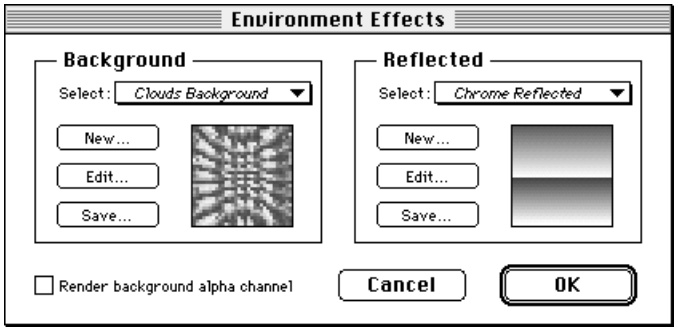
The Environment Effects command allows you to use various colors, gradations, or textures for backgrounds and reflective information in a rendering. This command is available whenever a model is open.

The default background environment for new models is a simple white color; the default reflective environment is a sky using dark colors. You may change the environment settings at any time. The specific environments loaded into a model are saved as part of the model document.

New environments that you create may also be saved as separate files for use with other models. This is done by clicking on the Save button in the Environment Options dialog. The currently-loaded environment — either background or reflected, whichever Save button is clicked — is saved as an environment file.

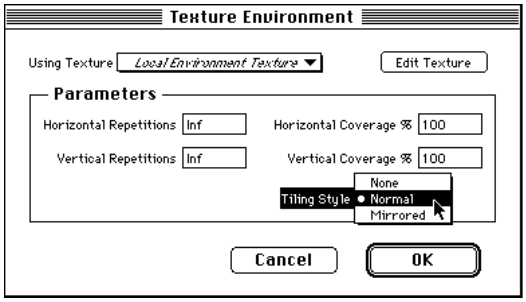
When an environment is loaded into a model, it is wrapped around the model on an imaginary sphere surrounding all objects, a sphere much like the one used to position directional light sources. The environment cannot be selected directly like an object, but may be managed through the Environment Options command and its various levels of dialogs.

Environments add a surrounding reflection and / or background to the scene. They also provide objects some type of context in the model. By using an environment instead of additional comprehensive modeling, the number of constructed objects or primitives to be rendered can be minimized or even reduced.



When the Environment command is called, a dialog box opens to select and load one or more environment files and specify how they are to be used. The dialog is split into Background on the left and Reflected on the right, allowing you to use one environment for the background and another for the reflections that appear in shiny surfaces. Each half of this dialog works the same.

You should note that texture-based environments will often require different scale and mapping parameters for either of the two sides. Backgrounds using textures with maps are usually scaled smaller. The scale for a texture-based environment is adjusted through the Edit button in the dialog. You should note that the horizontal and vertical coverage fields are expressed as percentages, i.e. 100% scales the map to cover the entire environment sphere once in the designated direction, a setting of 33% scales the map to cover the sphere three times in the chosen direction. The images appearing in the Environment Options dialog display how a portion of the environment looks as it is mapped onto the sphere. Use them to set the mapping scale larger or smaller and the tiling pattern as needed.



You will find it useful to load some of the environments included with Vision 3d and see the scale settings used for them, then adjust your own environment creations in a similar manner.

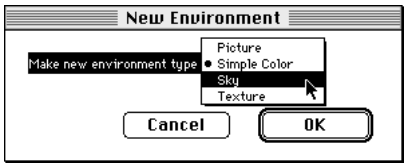
LOADING ENVIRONMENTS

An environment may be loaded from the “Environments” folder using the Select field’s pull-down menu. Available environments found in the folder at the time the application is launched appear in the menu. Chose the desired environment by highlighting its name on the list, then release the mouse button. The selected environment will load into the model.

If the environment you want to use is not in the Environments folder at startup, you cannot load it into the model. You must quit the application, move the desired environment into the folder and re-start. Unlike textures, no other way is provided to access environment-type files outside the specified folder.

NEW ENVIRONMENTS

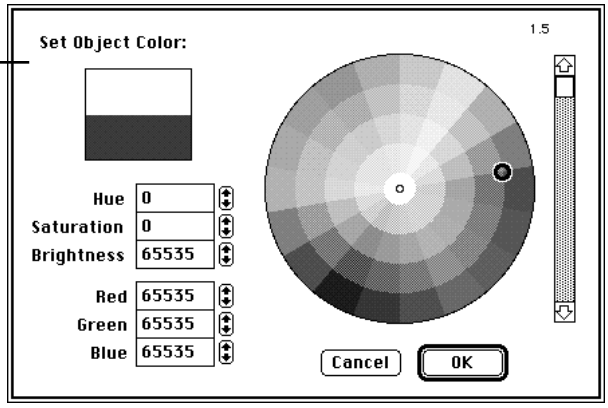
Vision 3d supports four types of environments: 1) Simple Color, 2) Sky, 3) Texture, and 4) Picture. Clicking on the New button opens a dialog to choose the type of environment you wish to create.



Then, based on the type of environment, a custom secondary dialog is opened to construct the environment type chosen. The four dialogs that open are shown in the following illustrations on this page and the next.

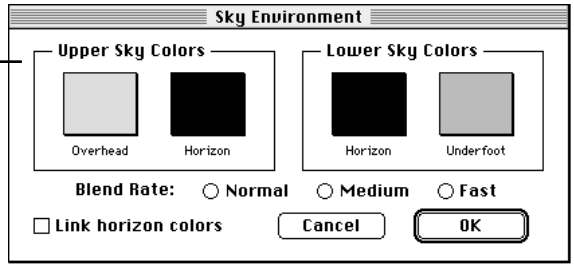
Simple color environment dialog

(A.K.A. the Apple Color Picker™)



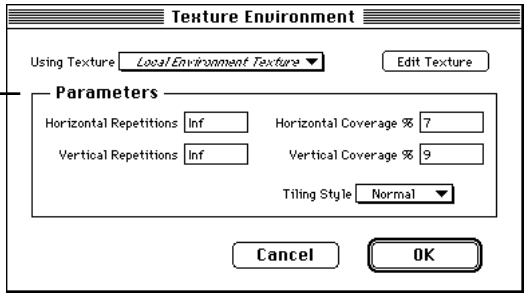
Sky environment dialog

Allows you to pick four colors that create a graduated color for sky and ground plane



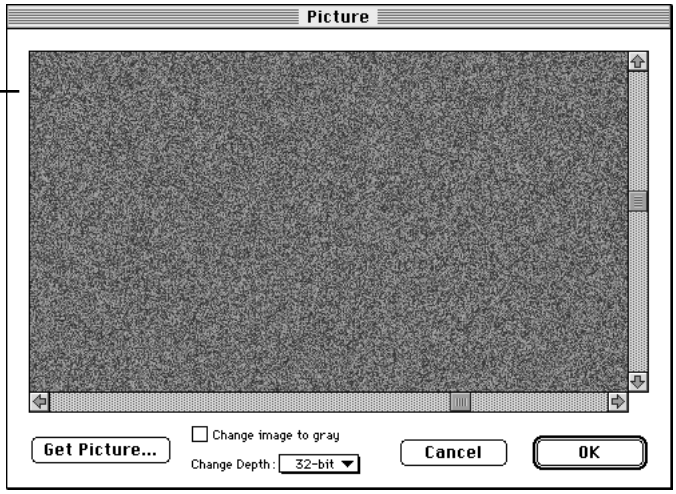
Texture environment dialog

This dialog box appears for environments using a texture to set scale and tiling



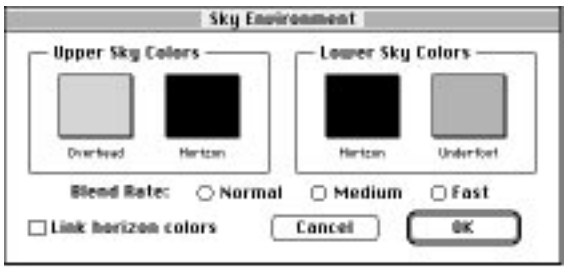
Picture environment dialog

This dialog box appears to select the PICT image you wish to use as a Picture environment

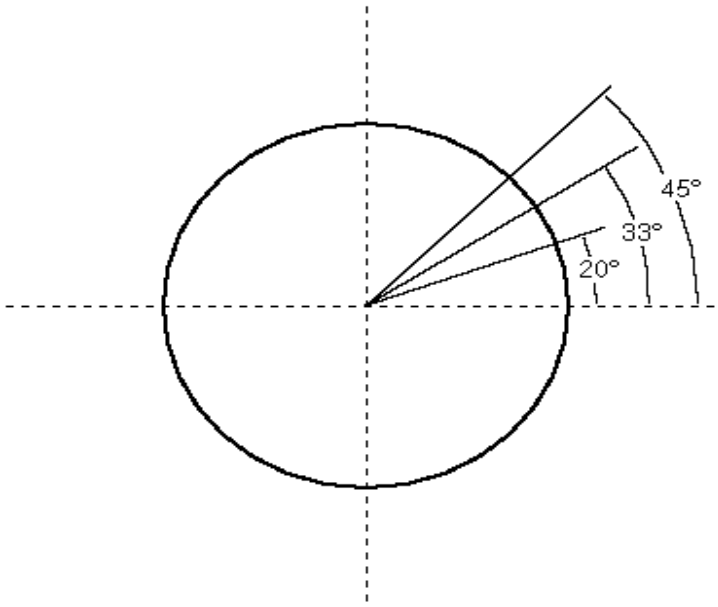


SKY ENVIRONMENTS

The Sky environment has been given an adjustable blend rate for better results in various situations. This is controlled by radio buttons in the environment's dialog box. An adjustable blend rate allows you to tailor the gradation to better match the lens angle of the view being rendered.



Normal blending uses a linear gradation from the horizon color to the overhead and underfoot colors. The transition from horizon color to sky color is complete at 45° above or below the horizon line. The Medium setting uses an accelerated or "skewed" linear gradation to complete the horizon color's transition at 33° above or below the horizon line. The Fast setting uses a logarithmic gradation to complete the color transition at 20° above or below the horizon line.

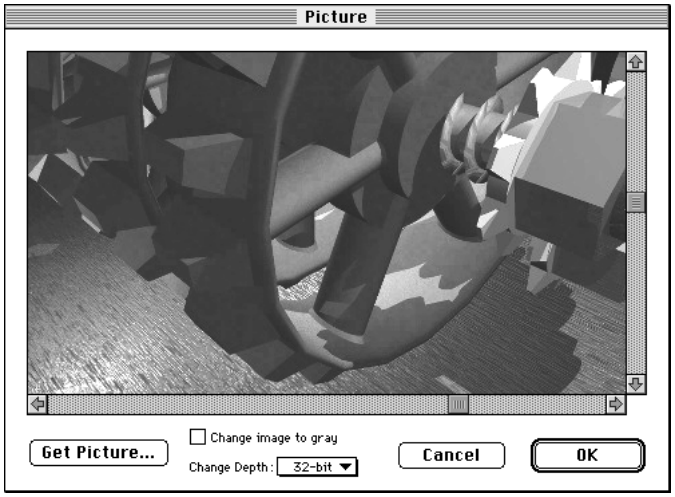


PICTURE ENVIRONMENTS

The Picture class of environments has been provided to support using background pictures for rendered image. You can prepare a PICT image to be used as a background through the Environment Options command.

When “Picture” is chosen as the environment type and the “OK” button is clicked, the Picture dialog box is presented to choose and prepare the PICT you wish to use as a background.

You may recognize this dialog as the same one used for loading PICT images for surface maps in textures. As before, the option for changing color depth is also available in this context of the use of this dialog.



The background picture is positioned and sized to fit the active view — whether in a modeling window or a camera window — from which

the snapshot is taken. If the active view is switched, the background picture will automatically re-size to match the active view's extents.

You must make sure the active view has the same proportions or aspect ratio as the PICT being used, or it is possible to distort the background picture. This is something you should watch while preparing the final rendering. If you define a snapshot smaller than the active view, the background picture is also cropped and you will see only that part of the background picture that is visible within the rendered image.

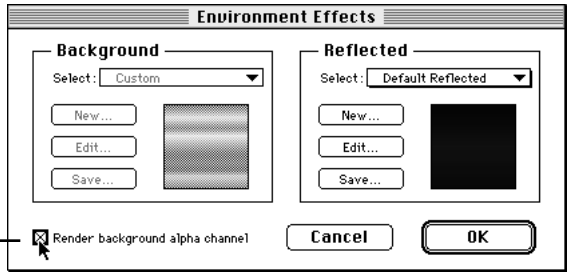
ALPHA CHANNEL SUPPORT

Vision 3d provides an option to render an Alpha channel mask of the background behind your scene. Enabling this option, which appears as a check box in the Environment Effects dialog, creates a mask that can make the background transparent if supported by the application in which you will be using the rendered image. This feature is available only if the rendering is saved in full-color (32-bit) PICT format.

The alpha channel is another eight-bit channel in the 32-bit color image. Eight bits each for the red, green and blue components add up to only 24 bits, so the alpha channel completes the 32 bits of information contained in the 32-bit color format. You can think of it as a grayscale image that masks the background.

Environment Effects dialog

When alpha channel option is turned on, the background environment is disabled



The alpha channel allows easy placement of a rendering into a page layout document and facilitates the "compositing" of renderings with graphical elements from other software packages.

When the alpha channel option is enabled, the background areas of the image are automatically identified during rendering. It doesn't

matter what the background color is, because the background will be masked to be transparent. Note that enabling the alpha channel disables the background environment. The background of the image renders as black with the grayscale mask in a separate channel to make the black area transparent.

To use the alpha channel option, select the Environment Effects command under the Rendering menu. Click on the check box in the lower left corner of the dialog window. The background environment side of the dialog grays out so it cannot be used, even if an environment has been previously loaded there. Click on the OK button, then render the scene as usual.

NOTE: *To realize the potential of alpha channel support, the finished rendering with background mask must be used in a software package that supports the alpha channel. Alternately, the mask must be converted to a clipping path in an image processing package like Adobe® Photoshop™ for use in packages that do not offer direct alpha channel support. Consult the user manual of the software package in which you plan to use your rendering with background mask to determine whether alpha channel support is present. And if so, how to use it.*

Make Optimal Palette command (Edit menu)

The Make Optimal Palette command generates an alternate set of colors that more closely match the colors used in a specific image. This command is available when a pixel-based color image is present in the active window.

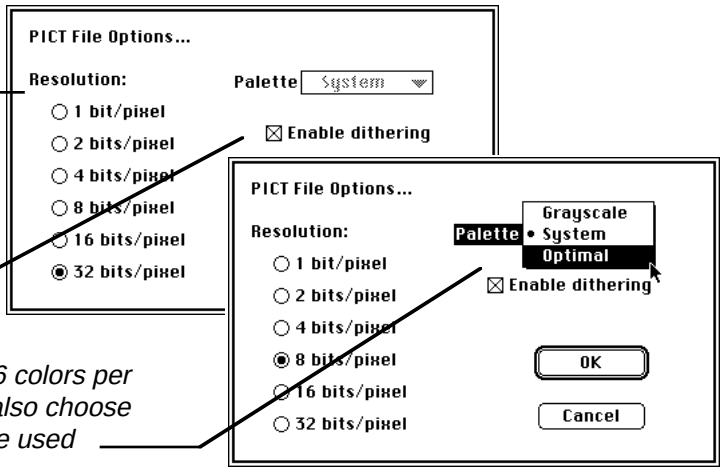
Note: *Under System 6.0, only color palettes are optimizable. Gray scale palettes continue to use the standard system palette. Under System 7.0, both color and grayscale palettes may be optimized.*

Calculating and attaching an optimal palette to an image allows it to be displayed on monitors of reduced color depth with less dithering. Since the colors used to display the optimized image are more closely aligned with the colors the image actually requires, the effect is a cleaner, truer color display of that image. The optimal palette is present only while the target image is open in the active window. Closing the image removes the optimized palette from the System.

When saving images, you can choose the color depth

…and whether the image will be "dither-able" when using QuickDraw to display it.

For color depths of 256 colors per pixel or less, you can also choose the type of color palette used



You may permanently attach an optimized palette to an image as part of the save operation. This is discussed in chapter 1 under the Save As command. Images saved with an optimized color palette can use that palette when they are read into Vision 3d again or in other applications that support this feature. The mechanism that allows this process to work across applications resides in the System software and 32-Bit Color QuickDraw. An image with an optimized color palette read into software that uses QuickDraw to display its images can have the System replace the standard system color palette with the attached optimized palette.

Modify Image command (Edit menu)

The Modify Image command allows the application of filters to an image. This command is available when a window containing an image is the active window. The image may be a completed rendering or one from another source read into Vision 3d using the Open command in the File menu.

Vision 3d provides two built-in filters: Gaussian Blur and Median Filter. With these filters, you can conveniently clean up visual noise or lightly blur an image.

THIRD-PARTY FILTERS

In addition, Vision 3d supports extension filters designed by Strata Inc. or third-party developers specifically for Vision 3d. Once installed, extension filters are accessed by this command. All filter commands operate on the entire image. Filters that have dialog boxes associated with them appear with an ellipsis (…) after their name in the sub-menu.

Each time a filter is chosen that requires parameters, you must re-enter the values.

Vision 3d also supports Adobe Photoshop's plug-in filters sold by third-parties. To use, they must be copied into the Extensions folder within StrataVision 3d's application folder before program startup. Under system 7, you can use aliases in the Extensions folder so you do not have to have a completed set of duplicate filters in two places on your hard-disk drive.

Choosing the Modify Image command displays a sub-menu, from which you can open specific filters for processing the image in the active window.

USING FILTERS WITH ANIMATIONS

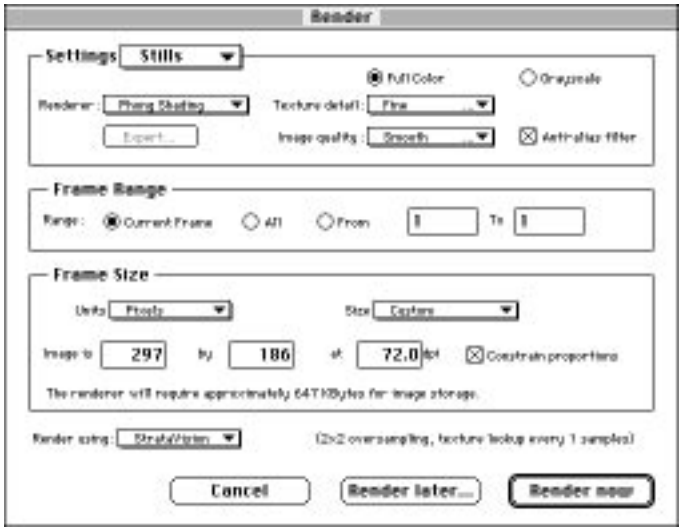
Both still images and animations can be post-processed. Both PICS and QuickTime Movie animation formats are supported. This feature may be used on animations from any source, not just Vision 3d.

When using the Modify Image command, if a filter is applied to a single image, the process proceeds in a straightforward manner. If the active window contains an animation (in either QT Movie or PICS format), a dialog appears asking the range of frames within the animation to which you wish the filter effect applied. The dialog always opens with only the current frame specified. You can change the dialog as needed before clicking on the “OK” button or pushing the Return key.

Render command (Rendering menu, ⌘ Y)

The Render command initiates a rendering. Render is available whenever a model, shape, or camera window is active. Selecting the Render command opens a dialog box allowing you to choose the renderer, image quality, set the frame or number of frames to be rendered, and the image size and resolution.

Clicking “Render Now” or pushing the Return key initiates a snapshot of the model from the currently active view. Alternately, you can choose the Render Later option to save a suspended rendering directly from this dialog.



When the Render command is selected directly from the Rendering menu, the currently active view is used for the rendering, as well as for size and proportion. The view's focal length setting also becomes the perspective used for the rendered image.

Dimensions for a snapshot can be adjusted directly in this dialog. They may be expressed in pixels, inches or centimeters. This is chosen through the Units field pop-up menu. You may set the resolution as a function of the units chosen. This will be the finished image's resolution after the anti-aliasing interpolation has been performed, if used.

You may choose from a set of pre-defined sizes or enter your own custom size for the rendered image. Based on the size and resolution settings, a field at the bottom of the dialog provides information on the effective resolution (including anti-aliasing, if turned on), and how much disk space will be required while the image is rendering.

Note that the size requirement shown is the absolute minimum, and may be more. You must have at least this much space free, either in RAM or for the virtual memory cache on your hard disk. Depending on the format chosen for saving the image when the rendering is complete, the saved file may be much smaller.

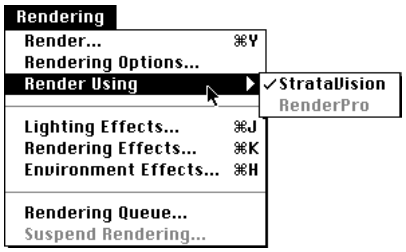
NOTE: *You may also access this dialog by holding down the Shift key while using the Camera tool to define a snapshot. This allows you to crop the scene being rendered, while also accessing the numerical control for image size, resolution, etc. provided by this dialog. When using the drag-define method with the Camera tool, adjustments made to the image size in the dialog apply to the area defined by the drag marquee.*

Render Using command (Rendering menu)

Use this command to select the source for rendering images. Vision 3d defaults to using its own built-in rendering algorithms each time it is run. To use an alternate rendering "engine" or application, you must select it from this command's sub-menu. Only one can be chosen at a time.

Only those rendering alternatives specifically supported by Vision 3d appear in the sub-menu. Some choices may communicate transparently with Vision 3d; returning the finished rendered image back to the application for displaying or saving. Others may require Vision 3d to create a separate suspended rendering or export file that must be

manually opened in the external rendering application. The specific process for using another renderer is explained in the documentation that should accompany that rendering solution.



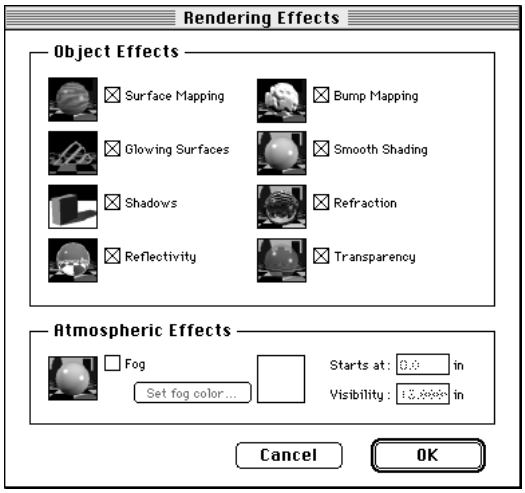
The choice of built-in or alternate renderers can also be made through the Render command's dialog box. A pull-down menu duplicates the "Render Using" command in that dialog box. It shows the currently-selected rendering source and allows you to change it, if needed, at that level too.

Render using pull-down menu

Rendering Effects command (Rendering menu, ⌘ K)

The Rendering Effects command is used to turn the rendering effects on or off globally. This command is available at all times.

Choosing this command opens the Rendering Effects dialog. This is where you enable or disable the effects you wish to see in a rendering. Each time Vision 3d is launched, all the rendering effects are turned on, except Fog. This is the default configuration.



You must consciously open this dialog and choose to turn off the effects supported in Vision 3d. Turning off effects that you do not need or want in a rendered image or animation can improve the rendering performance.

The effects available in this dialog are as follows:

- **Surface Mapping:** This field enables Vision's mapping feature. Two types of mapping are supported in Vision 3d: Surface and Solid. Both types have their strengths and weaknesses. Surface mapping is the process of using two-dimensional PICT images to alter the surface of objects. Surface mapping is the most usable and flexible, and often produces the most realistic images because the maps are usually taken from real objects. Solid textures use a mathematical formula to generate a pattern that is volume dependent. They must be compiled

externally before use in Vision 3d, but are more memory efficient and generally render faster. Some solid textures are shipped in the Vision 3d product. More solid textures are available from Strata Inc. in the form of commercial libraries.
Surface mapping is available for use with Phong, Raytracing, and RayPainting.

- **Glowing Surfaces:** Objects with this attribute have a luminescence greater than the light shining on them. This effect is often used to give an object or group of objects a neon-like appearance. The actual amount of glow for each object must be set in the texture applied to that object.
  Glowing surfaces is available with the Gouraud, Phong, Raytracer, and RayPainter algorithms.
- **Shadows:** This effect enables calculating shadows in the rendered image. All objects cast shadows unless an object's shadow is specifically turned off with the "Passes light" field in the texture applied to the surface of that object. Lights may also be given the option to not cast shadows through the Adjust Object dialog for the light source.
  The shadows rendering effect is available with Raytracing and RayPainting.
- **Reflectivity:** Objects with a reflectivity component in their surface produce a mirror-like appearance. The degree of reflectivity is variable with a specific object's texture settings. Enabling this rendering effect performs the calculations necessary to show reflections in reflective surfaces.
  Reflectivity is available with the Phong (limited to environments), Raytracer, and RayPainter algorithms.
- **Bump mapping:** This effect allows any bump-type surface maps present in textures to influence the appearance of object surfaces. Turning this effect off disables all bump maps throughout a model during rendering.
  The bump mapping effect is available with Phong, Raytracing, the RayPainter.
- **Smooth Shading:** This effect determines whether object surfaces are rendered smooth or faceted. Most curved surfaces are modeled as a collection of polygons. In the lower level rendering methods, objects will appear faceted. Faceting is the result of sudden changes in light intensity from one polygon to the next, represented in a rendered image as a change in color value.

The more sophisticated renderers can obtain a smoother surface appearance on objects by averaging the lighting change across curved surfaces, but only if the object is a primitive or its surface has been "smoothed," i.e. defined by a triangulated mesh with vertex normals present.

The smooth surfaces effect is available with the Gouraud, Phong, Raytracer, and RayPainter algorithms.

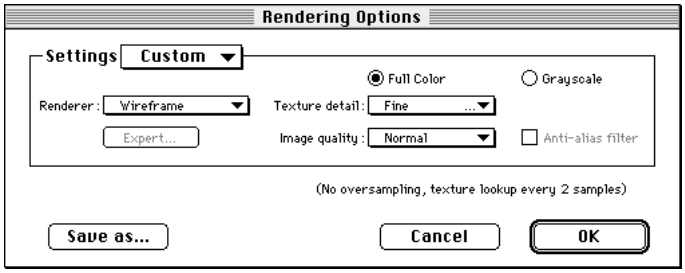
- **Refraction:** Light shining through transparent objects is "bent" by the object's material. This phenomenon, called refraction, is observed often in the real world. As a transparent object is being rendered, its surface must be modified to include contributions from light in back of the object. In most cases, this light is reflected from surfaces of other objects. Some of Vision's renderers can take refraction into account, depending on the lighting and placement of images behind them. This option adds refraction into the imaging calculations. Refraction is available with the Raytracer and RayPainter algorithms.
- **Transparency:** Objects with this attribute have a crystalline or glass-like appearance. The degree of transparency varies with a specific object's texture settings.

  The Wireframe, Hidden Line and Quick Shading algorithms do not support transparency. Flat shading, Gouraud and Phong rendering methods implement transparency using a "screen door" effect to see through an object's surface to those behind it. The amount of transparency is fixed in levels of zero, 25%, 50%, 75% and 100%. Ray-tracing and RayPainting can render transparent objects in fully variable amounts corresponding to the values in the transparency field for the surface on each object.

  Transparency is available with the Flat shading, Gouraud, Phong, Raytracer, and RayPainter algorithms.
- **Fog:** This effect enables a fog-like appearance in the rendered image. While not a true volumetric effect based on spatial density, it does simulate the effects of fog based on distance from the viewing plane. Fog is available with Flat Shaded, Gouraud, Phong, Raytracing, and RayPainting.
- **Fog Color, Starting Distance, and Visibility:** When the fog option is enabled, secondary fields become active to set the color of the fog, the distance from the viewing position when it starts in the scene and how dense the fog appears. There are no default or recommended settings for these fields.

Rendering Options command (Rendering menu)

The Rendering Options command is used to set (and optionally save) the parameters for a rendering. This command is available whenever a model is open. In this dialog, you can chose the rendering method or, based on the setting chosen, Vision 3d can select the most appropriate rendering algorithm to accomplish the task.



Choosing this command opens a dialog to select options desired for the next rendering you initiate. While this dialog is open, you may also save various combinations of options by name for use as pre-defined settings with the pull-down menu in the Tool palette or in this dialog and the Render dialog.

The Rendering Options dialog includes the following fields:

GRAYSCALE RENDERING

The grayscale option renders the model in 256 shades of gray. This option is useful for rendering images that must reproduce well using a B&W output device, such as a laser printer or high-resolution typesetter, or for use in B&W printing applications.

Selecting gray scale forces both the screen image and the internal image to represent colors in shades of gray, even if the model has multiple-color attributes assigned to objects. The resulting image is saved in 8-bits per pixel format

FULL-COLOR RENDERING

Selecting full color renders images in a format compatible with 32-bit Color QuickDraw. The image is rendered in full 32-bit color, even if the monitor displaying the rendering window is not capable of showing that much color depth. Color QuickDraw provides a standard way of

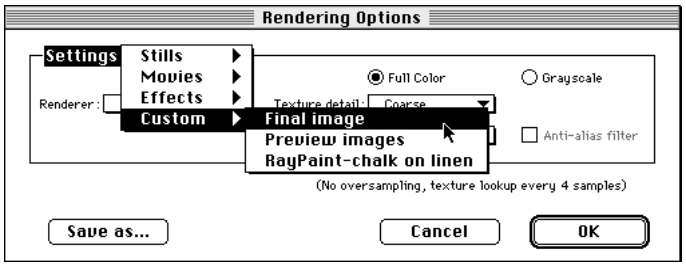
handling images, no matter what device is used to reproduce or display it — from B&W laser printers to color film recorders, from gray-scale monitors to high resolution video film.

The 32-bit Color QuickDraw format supports B&W, 8-bit, 16-bit and 24-bit color devices, allowing full-color images to be displayed or output in one color, 256 colors, thousands or millions of colors. Approximation of an image's actual colors at lower resolutions is accomplished via dithering.

The rendering window displays the image at the best capability of the monitor. To see the full range of colors contained in a 32-bit color image at the time of rendering, you will need a full color monitor. The resulting image is saved in 32-bits per pixel format.

SETTINGS

This field allows you to set the current options and values in the dialog to a previously-defined configuration that was saved previously through the Save As button. Previously-defined rendering options are saved between sessions in the Preferences file and are available to all models. However, the Preferences file is not updated with any newly-saved rendering options until Vision 3d is quit. If Vision 3d is exited other than through the Quit command, the newly-saved rendering options will not be available at the next session.

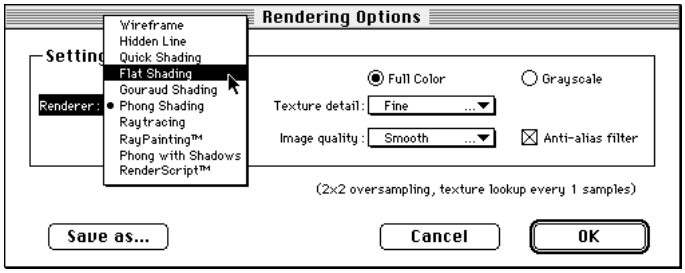


The pre-defined options list appears in the pull-down menu for selection. Picking an entry affects all the check boxes, numbers and settings in the Rendering Options dialog, including those in the Raytracing and RayPainting esoterica dialogs. Thus, whatever values are in these fields at the time a configuration is saved will be re-entered into those same fields upon selecting the choice again from the Settings pull-down menu.

The list of pre-defined rendering options is also available in the Tool palette and the Render dialog.

RENDERER SELECTION

Clicking on this field displays a pull-down menu. Vision 3d uses the rendering mode chosen even when the rendered image desired doesn't require it or, conversely, when a higher level of rendering may actually be preferred.



The following choices with their rendering capability appear in the menu:

- **Wireframe: With Wireframe selected, only the outlines of objects are drawn, without indication of surfaces. This is one of the methods** Vision 3d uses to represent objects in the modeling windows, but when chosen here for rendering, the Anti-aliasing capability is also available.
- **Hidden Line: This setting produces a rendering of wireframe objects, but with surfaces implied by the removal of hidden lines. Again, this is a display method used for the modeling windows, but is also available here.**
- **Quick Shaded: This setting renders an image with simple shaded objects. In addition to showing object** surfaces, lighting direction and the resulting shading of objects is calculated. This is a simple rendering method using only one lighting calculation per polygon.
- **Flat Shading:** Selecting Flat Shading renders with solid surfaces and colors. Transparency is supported using a screen door appearance in increments of zero, 25%, 50%, 75% and 100% transparency.
- **Gouraud**: Gouraud is a shading-oriented rendering algorithm that refines how an objects' surfaces are shaded. This rendering mode can execute the smooth shading option if the object is either a primitive

type or defined as a triangulated polygon mesh with appropriate vertex normals.

- **Phong Shading:** Phong rendering is a light-oriented algorithm, with the capability to render smooth surfaces, reflectivity (environments only), transparency, and surface maps.
- **Ray tracing:** This rendering method calculates how rays of light react to object surfaces within the model before reaching the viewer's eye. Raytracing creates enhanced photo-realistic images with true reflectivity, shadows, refraction, transparency, etc.
- **RayPainting:** This rendering method provides an artistic flair that diverges from the photo-realistic intentions of other high-level renderers. RayPainting creates images with 3-D brush strokes on textured backgrounds.
- **Phong with Shadows:** As the name implies, this rendering method produces a Phong rendering with shadows. However, this renderer will only produce shadows if "Shadows" is checked in the Rendering Effects dialog box.
- **RenderScript™.** This output of this rendering method is an EPS (Encapsulated PostScript™) file with bézier regions. This rendering technique can be used to create surrealistic images which can be scaled to any size.

The choices discussed in this menu are presented in lowest-to-highest order of rendering sophistication. Depending on the rendering method chosen, a finished image can take from just a couple of seconds to several hours to render. You must balance the need for realism with the time the higher-quality rendering algorithms require to perform their work.

IMAGE QUALITY

The Image Quality field sets the rendering detail for the desired image quality. Predictably, each increase in the level of image quality increases the quality of the rendered image and also requires longer to render. The options are represented in this dialog with descriptive titles.

- **Anti-aliasing:** The "stair-stepped" appearance visible on diagonal edges and lines can be minimized or eliminated with this effect turned on. This is done by choosing either the Smooth or Extra Smooth image quality setting.
  Anti-aliasing is independent of any specific rendering algorithm. It may be used with any set of rendering effects and/or rendering algorithm. The level of anti-aliasing chosen depends on the need.

The Smooth setting is recommend for all but the most critical imaging situations.

- **Anti-aliasing Filter:** When the Smooth or Extra Smooth option is used in the Image Quality field, the “Anti-alias filter” checkbox becomes active. This enables an additional pixel averaging step that further enhances the appearance of a rendered image.

TEXTURE DETAIL

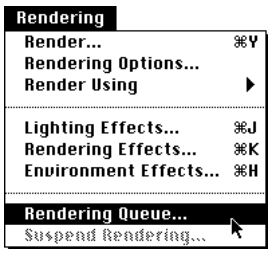
Texture detail is set independently of the rendering algorithm or the image quality. Of course, not all rendering methods can use the information provided by textures. As the level of detail increases, the quality of the surface detail increases, with a corresponding increase in rendering time.

This field sets the rate at which surface maps are sampled. A setting of Fine means the maps are sampled every pixel. A setting of Normal samples every other pixel; Medium samples every third pixel; and Coarse samples every fourth pixel. The pixels in between the sampled pixels are extrapolated and filled in by Vision 3d.

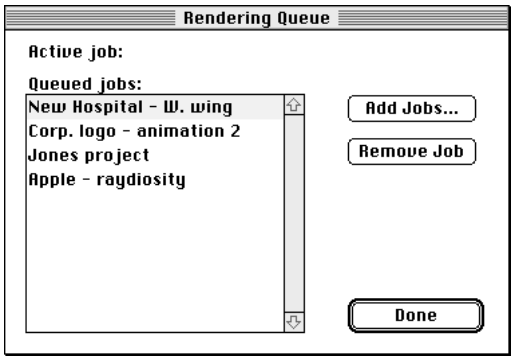
The sampling rate is a global setting, affecting all objects in the scene being rendered. The sampling rate chosen depends on the desired detail of the maps. It involves the usual trade off between image quality and rendering speed. The lower the detail, the fewer times a map must be sampled during the rendering process. Typically, you would use the lower detail settings for preview renderings during the modeling and scene-assembly phase, then use a sampling rate of one or two (fine or Normal) for the finished image.

Rendering Queue command (Rendering menu)

This command allows you to add suspended renderings to or remove them from the rendering queue. The Rendering Queue command is available all the time.



A queue is established when one or more suspended renderings are added to the job list in the Rendering Queue dialog. Only one rendering is active at a time. The order in which the jobs are added to the queue determines the order they are rendered. Clicking on the Done button begins the loading and rendering of the first rendering in the queue.



When a queued rendering is closed, either because it has been completed and been saved or it has been terminated by clicking on the close box, the next rendering in the queue (if any) loads and begins rendering.

If Vision 3d is quit while renderings are in the queue, they are deleted from the queue. Vision 3d does not remember the queuing of renderings between sessions. An alert is displayed announcing that renderings are still present in the queue, and asking if you are sure you want to quit.

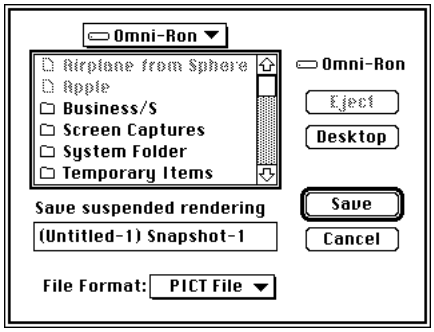
If you want to render more than one image at a time, you must open them manually. The rendering queue has no provision to support

multiple renderings at one time. Remember, with more than one rendering in progress at a time, you must send them all to the background so they will share CPU time. Otherwise, the one in the active window (foremost on the desktop) takes priority and all the CPU time…just like using the rendering queue.

Suspend Rendering command (Rendering menu)

The Suspend Rendering command allows you to save a rendering in progress, which can then be restarted at a later time. Suspend Rendering is available when a rendering window is the active window and the rendering is in progress.

All the model data, as well as the rendered image at its current state of completion, are saved as a self-contained file. You do not need to have the original model open to restart the suspended rendering.



Suspended renderings are re-started using the Open command in the File menu. Upon restarting a suspended rendering, its rendering window re-opens to monitor the rendering's progress just as before.

Suspended renderings are useful when you need to schedule more involved Vision 3d projects around other tasks requiring the full attention of the computer. Suspended renderings may be restarted, allowed to render for a time, suspended again, then restarted again…as time permits and as often as necessary to complete the rendering project.

Additionally, the Rendering Queue command is available for queuing several renderings to load, render, and save automatically. This command works only with suspended renderings.

GLOSSARY

- **Ambient light.** Light that is present in the environment. It has no focus or direction.
- **Anti-aliasing.** A mechanism to prevent or remove the jagged appearance of diagonal lines or edges in an image. Anti-aliasing can be achieved in rendered images by averaging adjacent pixels with sharp variations in color and brightness or by increasing the resolution of the image to meet or exceed the resolution of the device displaying or printing it.
- **Bézier splines**. Lines that are curved or capable of being curved. The curve of the line is defined by vertex points.
- **Bitmap.** Also known as a pixel image. An image composed of pixels. May be any resolution or color depth. Rendered images may be saved as a bitmap file in either PICT or TIFF format. Surface maps are contained in textures as bitmaps.
- **Bounding Box.** An imaginary rectangle that exactly encloses the geometry of an object or group of objects. When an object is selected, handles appear at the corners and midpoints of the sides of this rectangle. When an object is manipulated using one of the object-handling tools, its bounding box appears to interact with the tool and provide visual feedback during the tool operation.
- **CAD.** An acronym that stands for Computer-Aided Design.
- **Click.** The act of positioning a tool cursor in a view, and usually on an object, then pressing the mouse button. This event initiates a process or operation related to the tool's function.
- **Click-Drag.** The process of clicking the mouse button and, while holding the mouse button down, dragging the cursor in the view. For example, this procedure would be used to move an object from one location to another in a view.
- **Cursor.** An icon indicating the current tool selected and / or the current focus of the program. Vision 3d uses different cursors to indicate the selected tool.
- **Database.** The area of memory within the program while it is running that is set aside to keep track of objects within a model.
- **Default.** A parameter or setting pre-defined in the program which may be changed by the user.

- **Diffuse Reflection.** That component of the light reflecting from a surface caused by its dull or matte nature. Dull or Matte surfaces reflect the light striking them in random angles over a large area; giving the surface an equally-bright appearance from a wide range of viewing positions.
- **Double-click.** The act of positioning the cursor on an object in the active view or an icon in the interface, then pressing and releasing the mouse button twice in rapid succession.
- **Extrusion.** A method of creating a 3-D object using a 2-D template; giving depth to the 2-D shape.
- **Flat Shading.** Shading by filling each facet of a surface with a single color.
- **Fractals.** Used to simulate the random and/or chaotic characteristics of natural phenomena. Fractal geometry provides a mathematically produced model for simulating complex forms found in nature.
- **Gouraud Shading.** A rendering method that generates a smooth-appearing surface similar to the Phong shading method, but uses normals attached to the vertices for its shading calculations.
- **Group.** A collection of objects that act as one.
- **Hierarchical Object.** Creating an object shape in a separate work space, which when instantiated in the model retain a link to the shape. Changes made to the shape are automatically applied to the instantiations.
- **Instance**. A replica of a shape. Each time a shape is inserted into a window, an *instance* of that shape is actually inserted. Any changes made later to the original shape are reflected in each instance, or occurrence, of the shape. Changes made to the instance, however, do not in any way affect the root shape.
- **Lathing.** A method of creating a 3-D object by revolving a 2-D profile about a designated axis.
- **Marquee.** A rectangular dotted box created by dragging the cursor. Marquees appear in the programs windows or views to provide visual feedback during various tool operations.
- **Mesh Surface.** A surface that has common vertex points between adjacent polygons. Mesh surfaces are typically used to define complex forms. The surface also provides smoothing information for the rendering algorithms.

- **Normals.** The mathematical value that indicates the direction a surface is facing in 3-D space. Normals may be attached to individual polygons or to the vertex points that define the polygons.
- **Orthographic.** A method of displaying objects in a view where parallel lines do not converge. Orthographic projections are used to compare object's absolute dimensions, without the confusion of relative distance from the view.
- **Parent/Child Link.** An internal connection between two objects, in which the child is linked to the parent for constraint purposes. Various uses exist in different applications for linked objects, but are usually related to animation.
- **Perspective.** A depth cue available in Vision 3d in which parallel lines converge to align with a designated vanishing point. Perspective is a product of several parameters in the displayed view: Lens focal length, camera size, and distance from objects.
- **Phong Shading.** A rendering method that creates a smooth surface by blending the boundaries between adjacent polygons of a mesh surface. Shading calculations are based on the normals attached at the center of each polygon. Phong Shading supports surface and volumetric mapping.
- **Pixel.** An acronym that stands for Picture Element. It is the smallest component which makes up the display on a computer monitor. Each dot on the screen is a pixel. Many images displayed on the screen are likewise stored in a pixel form that is mapped to the screen pixels for viewing.
- **Point light**. A local source of illumination that shines in all directions from a single point.
- **Polygonal.** A closed plane bounded by three or more line segments.
- **Primitives.** Basic geometric elements from which complex objects can be built.
- **Radiosity**. The process of determining how light is transferred between surfaces based on their color and proximity. Radiosity is capable of producing remarkable realism.
- **Raytracing.** A rendering algorithm which simulates the physical and optical properties of light rays as they reflect off objects in a 3-D model. This method of rendering typically traces rays of light backward from the imaging plane toward the light sources.

- **Refraction.** The change in direction of light as it passes from one transparent material to another. This causes an apparent shift in the image showing through the transparent material.
- **Rendering**. A visual representation of the model.
- **Specular Reflection.** That component of the light reflecting from a surface caused by its shiny or glossy nature. Shiny surfaces reflect light striking them in clearly defined angles of incidence; resulting in "hot spots" corresponding to the direction of the light sources providing the illumination.
- **Surface Mapping.** A process in which a PICT image is used to define an object's surface properties. A separate map is used to define color, texture, reflectivity, transparency.
- **Spot light**. A local source of illumination which shines in only one direction
- **Sweep.** A method of creating a 3-D object from a 2-D template using a combination of lathe and extrude. The template is both pushed through space and revolved around an axis at the same time to create a geometrically complex shape. This type of modeling would be used to create the threads on a bolt, for example.
- **Vertex.** A point of intersection of two vectors or a point used to define a polygon.
- **Volumetric Mapping.** A process in which a map is defined as a 3-D volume, but which is only visible on the surface of an object. The object has the appearance of being carved out of the volume. Volumetric maps may also be used to define multiple surface characteristics.
- **Wireframe.** A fundamental rendering method that represents 3-D objects with connecting lines. No surfaces are shown, but may be implied if hidden lines are removed for the resulting image.
- **X-axis**. The horizontal axis which represents width.
- **Y-axis**. The vertical axis which represents height.
- **Z-axis**. The axis which represents depth.

APPENDIX A: BIBLIOGRAPHY

Here is some information on reference material pertaining to the rendering algorithms used in StrataVision 3d. The books in the list discuss the theory of three-dimensional graphics and explain many concepts pertaining to 3-D rendering. They should provide a good background in understanding how StrataVision 3d's rendering algorithms work. You should be aware that the material in these books can be very technical. You may also find that they are dated. The technology and software available in the field of three-dimensional modeling/rendering is progressing very rapidly.

- **"An Introduction to Ray Tracing,"** Academic Press © 1989. ISBN #0-12-286160-4. Edited by Andrew S. Glassner.
- **"Computer Graphics for Designers and Artists,"** Van Nostrand Reinhold Company Inc. © 1986. ISBN #0-442-24712-5. Written by Isaac V. Kerlow and Judson Rosebush.
- **"Computer Graphics - Systems & Concepts,"** Addison-Wesley Publishing Company © 1987. ISBN #0-20-114656-8. Written by Rod Salmon and Mel Slater.
- **"Computer Vision,"** Prentice-Hall, Inc. © 1982. ISBN #0-13-165316-4. Written by Dana H. Ballard and Christopher M. Brown.
- **"The Mathematical Structure of Raster graphics,"** Academic Press © 1989. ISBN #0-12-257960-7. Written by Eugene L. Fiume, University of Toronto, Ontario, Canada.
- **"Computer Graphics: Principles and Practice,"** Addison-Wesley Publishing Company © 1990. ISBN #0-201-12110-7. Written by James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes.
- **"A Progressive Refinement Approach to Fast Image Generation,"** COMPUTER GRAPHICS (SIGGRAPH '88 Conference Proceedings), Vol. 22, number 4, August 1988. ISBN #0-89791-275-6. Written by Michael F. Cohen, Shenchang Eric

Chen, John R. Wallace, and Donald P. Greenberg, Cornell University.

- **"A Ray Tracing Solution for Diffuse Inter-reflection,"** COMPUTER GRAPHICS (SIGGRAPH '88 Conference Proceedings), Vol. 22, number 4, August 1988. ISBN #0-89791-275-6. Written by Gregory J. Ward, Francis M. Rubinstein, and Robert D. Clear, Lawrence Berkeley Laboratory.

Of course, each of these books contains a large Bibliography that can point you to other books, publications or graduate-level theses on the subjects of computer modeling and three-dimensional rendering.

APPENDIX B:
TEXTURE SETTINGS FOR COMMON MATERIALS

	Ceramic	Paint	Paper	Plastic	Plexiglass	Rubber	Metal
Ambient Fraction	0.55	0.35	1.00	1.00	0.00	0.15	1.00
Bump Amplitude	1.00	1.00	1.00	1.00	1.00	1.00	1.00
Diffuse Fraction	0.60	0.50	1.00	1.00	0.20	0.70	0.00
Glow Factor	0.00	0.00	0.00	0.00	0.33	0.00	0.00
Reflectivity	0.25	0.15	0.00	0.15	0.15	0.00	0.30
Refraction Index	1.00	1.00	1.00	1.00	1.20	1.00	1.00
Specular Fraction	0.95	0.33	0.05	0.90	1.00	0.05	1.00
Transparency	0.00	0.00	0.00	0.00	1.00	0.00	0.00
First Smoothness	800	256	30	512	1024	6	900
Second Smoothness	0	0	0	0	0	0	6
Second Weight %	0	0	0	0	0	0	65
Enable shadows	yes	yes	yes	yes	yes	yes	yes
Link Specular / Diffuse	no	no	yes	yes	no	yes	yes
Prop. values scale maps	n / a	n / a	n / a	n / a	n / a	n / a	n / a

APPENDIX C:
REFRACTION: LIST OF MATERIALS

The following list of indexes of refraction was compiled and edited from the 63rd edition of the "Handbook or Chemistry and Physics" published by CRC press. This list is offered as a guide only in applying custom refractive indexes to textures which will be simulating these materials.

Minerals

Amber	1.455
Amethyst	1.544
Argonite	1.530
Azurite	1.730
Bentonite	1.757
Beryl	1.565
Borax	1.4466
Brucite	1.560
Calcite	1.658
Calomel	1.973
Celestite	1.621
Chlorite	1.570
Chromite	2.160
Diamond	2.4175
Dolomite	1.679
Emerald	1.576
Erythrite	1.626
Fluorite	1.433
Gypsum	1.519
Halite	1.544
Helvite	1.728
Hematite	3.220
Lazulite	1.604

Leucite	1.508
Magnetite	2.020
Moss Agate	1.540
Onyx Marble	1.486
Opal	1.450
Quartz	1.544
Rose Quartz	1.550
Rhodonite	1.711
Ruby	1.766
Sapphire	1.770
Serpentine	1.530
Smoky Quartz	1.550
Spinel	1.719
Sulfur	1.958
Talc	1.539
Topaz	1.606
Turquois	1.610
YAG Rhodochrosite	1.830
Zircon	1.923

Inorganic Compounds

Aluminum Chloride	2.700
Aluminum Oxide	1.665
Barium Oxide	1.980
Calcium Borate	1.540
Calcium Carbide	1.750
Calcium Sulfite	1.590
Copper Chloride	1.644
Common Glass	1.600
(ranges from 1.5 to 1.9, depending on composition)	
Ice	1.305
Iron Sulfate	1.802
Lead Nitrate	1.782

Lithium Bromide	1.784
Lithium Chloride	1.662
Lithium Oxide	1.644
Manganese Borate	1.617
Manganese Chloride	1.555
Mercury Cyanide	1.645
Mercury Iodide	2.748
Potassium Cyanide	1.410
Potassium Fluoride	1.352
Sodium Bromide	1.479
Sodium Cyanide	1.452
Sodium Nitrate	1.587
Sodium Phosphate	1.440
Strontium Fluoride	1.442
Strontium Oxide	1.870
Tin Iodide	2.106
Water	1.333
Zinc Bromide	1.545
Zinc Silicate	1.616

Organic Compounds

Methanol	1.326
Ethyl Ether	1.352
Acetone	1.357
Ethanol	1.359
Ethyl Acetate	1.370
Acetic Acid	1.370
Proprionic Acid	1.385
Butyric Acid	1.397
Sulfuric Acid	1.427
Ethelene Glycol	1.434
Chloroform	1.444
Fluorobenzene	1.463

Toluene	1.494
Benzene	1.498
Methyl Benzoate	1.515
Benzyl Alcohol	1.538

Realistic Gems

Another excellent source of information on the optical properties for various materials is “GEMSTONES of the World,” by Walter Schumann. ©English-language edition: Sterling Publishing Co. Inc., New York 10016. ISBN 0-8069-3089-6.

You should note, however, that in order for polygonal objects in StrataVision 3d to realistically render with the type of complex crystalline appearance of precious and semi-precious gems, they must contain an interior structure of transparent surfaces through which the light can interact and refract realistically. Additionally, you must pay special attention to the following points:

1. You must model the correct geometry for the gem(s) you wish to simulate, i.e. does the gem have a Cubic, Tetragonal, Hexagonal, Trigonal, Orthorhombic, Monoclinic, or Tricliniccystalline shape?
2. You must create and apply a texture that uses a second smoothness value corresponding to the internal crystalline structure. (This is definitely one of those trial and error type of projects.)
3. You must increase the value in the transparency recursion field of the Raytracer Esoterica dialog box (opened through the Rendering Options command’s dialog box) so that the multiple transparent surfaces will be fully noted and rendered by the raytracer. In the case of rendering realistic gems, the raytracer is definitely the algorithm of choice.

INDEX

Symbols

2-D OBJECTS
  extracting from B&W PICTs 2.34
2d SCULPTER
  changing the beginning vertex point 3.101
  converting a polygonal object to bézier splines 3.103
  description 3.94
  different menu bar 3.95
  explanation of the controls and tools 3.97
  how to use 3.97
  opening a window 3.95
  removing and adding vertex points 3.102
  Sculpt tool operation 3.105
  vertex editing 3.98
  working with bézier splines 3.103
3-D OBJECTS
  extracting from grayscale PICTs 2.34
3-D space (navigating in) 1.31
3d SCULPTER
  add vertex point tool 3.111
  description 3.110
  general vertex move tool 3.111
  horizontal move tool 3.111
  in/out Move tool 3.111
  Magnify tool 3.112
  perspective 3.110
  rotate tool 3.111
  vertical move tool 3.111
  window controls and modeler tools 3.109

A

About the interface 1.2
Absolute coordinates 1.35
Allocating application partition size 1.29
Alpha channel support 8.61
Animation extensions 7.71
ANIMATION MENU
  Allow Frame Skipping 7.76
  Cycle command 7.75
  Full-screen Playback command 7.79
  New Event Marker command 7.75
  Play Sound command 7.78
  Playback Controls sub-menu 7.75
  Playback Frame Rate command 7.80
  Playback from Memory command 7.77
  Re-animate command 7.81
  Reverse on Loop command 7.76
  Reverse Sequence command 7.82
  Show Animation Paths command 7.83
Animation palette 7.37
Animation windows (rendering and play back) 7.73
ANIMATIONS
  actions that are time-dependent 7.5
  Background Color command 7.80
  Bounce extension 7.72
  commands and controls 7.72
  Convert to Path extension 7.75
  copying and pasting frames 2.15
  event marker description 7.8, 7.39
  example of scripting simple actions 7.11
  frame differencing 7.63
  hardware limitations 7.2
  hierarchical scripting 7.10
  keyboard equivalents for animation window controls 7.79
  minimum requirements 7.7
  modifier keys that apply to event markers 7.40
  playing back partially-completed animations 8.15
  Point At extension 7.70
  QuickTime compressors, explanation 7.65
  QuickTime support 7.54
  rendering 7.30
  saving 2.12
  scripting (introduction) 7.1

selecting and moving event markers on the palette 7.39
visible motion paths 7.9
Anti-aliasing 8.22, 8.74
Application methods (textures) 5.54
Applying textures 5.54
Attenuation-setting for point lights 6.13

B

Background Color command 7.80
Bézier objects - modifying 3.116
Bézier objects vs. polygonals 3.55
Bézier spline text construction 3.80
Bézier surface patches vs. Polygonal meshes 3.113
Bézier to polygonal conversion 3.81
BOOLEAN
description 3.31
Intersection 3.42
operations 3.32
Subtraction 3.38
Union 3.33
Bounce 7.60
Bounce - carrying object motion into 7.62
Bounding boxes 3.46
Button bar 1.14, 3.91

C

Cache - setting location 1.27, 1.102
CAMERA OBJECT
adjusting frame size 3.175
adjusting parameters 3.173
Camera tool 3.87
Camera window - opening 3.178
Change Mapping dialog
description 5.55, 5.94
not available 5.55
Close command (File menu, ⌘ W) 1.82
Closing models 2.6
Color-setting for spot lights 6.12
COMMANDS
Actual Size (Windows menu) 3.161
Align (Modeling menu) 3.162
Close (File Menu, ⌘ W) 1.82
Copy (Edit Menu, ⌘ C) 1.84
Dither Image (Edit menu) 8.54
Duplicate 3.21
Duplicate (Edit menu, ⌘ D) 3.164
Edit Primitive (Modeling menu, B) 6.39
Edit Primitive (Modeling menu) 3.164
Environment Effects (Rendering menu, ⌘ H) 8.54
Group (Modeling menu, ⌘ G) 3.165
Hide Selected 3.166
Import (File menu, ⌘ I) 2.38
Lighting Effects (Rendering menu, J) 6.39
Lock (Modeling menu) 3.166
Make Optimal Palette (Edit menu) 8.62
Modify (Modeling menu) 3.169
Modify Image (Edit menu) 8.64
Mouse Filter 3.167
New (File menu, ⌘ N) 1.89
New Camera (Windows menu) 3.171
New Window (Windows menu, /) 1.90
Open (File menu, ⌘ O) 1.90, 2.41
Open As (File menu) 1.91, 2.42
Page Setup (File menu) 1.93
Palettes (Windows menu) 1.112
Paste (Edit menu, ⌘ V) 1.96
Print (File menu, ⌘ P) 1.104
Quit (File menu, ⌘ Q) 1.105
Re-center (Modeling menu) 3.179
Render (Rendering menu, ⌘ Y) 8.65
Render Using (Rendering menu) 8.66
Rendering Effects (Rendering menu, K) 8.68
Rendering Options (Rendering menu) 8.71
Rendering Queue (Rendering menu) 8.75
Replicate 3.21
Replicate (Edit menu, ⌘ R) 3.179
Revert to Saved (File menu) 1.106, 2.6

Save (File menu, ⌘ S) 1.107
Save A Copy As (File menu) 1.106
Save Window (Windows menu) 1.107
Select All (Selection sub-menu, ⌘A) 3.180
Select Next 3.180
Select None 3.181
Set Rulers (Edit menu) 3.181
Show All 3.182
Show/Hide Lighting Palette (Windows menu) 6.41
Show/Hide Tool Palette 3.182
Split View (Windows menu) 1.113
Suspend Rendering (Rendering menu) 8.77
Transform (Modeling menu, ⌘ T) 3.184
Undo/Redo (Edit menu, ⌘ Z) 1.113
Ungroup (Modeling menu, ⌘ U) 3.185
Unlink (Modeling menu) 3.185
Unlock (Modeling menu) 3.186
Commands and controls used in modeling operations 3.161
Compound objects 2.30
Convert to Path 7.75
Coordinate system(s) 1.34
Copy command (Edit menu, ⌘ C) 1.84
Creating new models 2.1
Customize menus command (Edit menu) 1.84
Cut command (Edit menu, ⌘ X) 1.85
Cycle command (Animation menu) 7.75

D

DEFAULTS
color or light 5.82
display method 1.21
foreground/background colors 1.24
lens perspective 1.21
window arrangement 1.23
Delete command (Edit menu) 1.86
Directional Light Source (applying gels) 5.85
Dither Image command (Edit menu) 8.55
“Drag and drop” explained 5.53
Drawing tools (2-D objects) 3.75
Duplicate command 3.20
DXF (file format) 2.26

E

EDIT MENU
Copy command (⌘ C) 1.84
Customize menus command 1.84
Cut command (⌘ X) 1.85
Delete command 1.86
Dither Image command 8.54
Duplicate command (⌘ D) 3.164
Make Optimal Palette command 8.63
Modify Image command 8.65
Paste command (⌘ V) 1.96
Preferences command 1.97
Replicate command (⌘ R) 3.179
Set Rulers command 3.181
Show/Hide Clipboard command 1.111
Show/Hide Rulers command 1.111
Undo/Redo command (⌘ Z) 1.113
Edit Primitive command 6.39
Enable Map Optimization 1.30, 1.103
Enable Projectors (checkbox) 6.40
ENVIRONMENTS
loading 8.59
picture 8.61
sky 8.61
using textures as 5.86
EPS (file format) 2.27
Esoterica dialog 8.39, 8.45
EXPORTING
DXF files 2.26
EPS files 2.27
general discussion 2.16, 2.24
RIB files 2.37
TIFF files 2.37
EXTENSIONS
explained 1.37, 3.2
EXTRUDE
description 3.122

how to use 3.122
making custom profiles 3.126
refractive textures 3.125
Extruding text - Tutorial 3.9

F

FACET
controlling density 3.129
description 3.127
grouping polygons 3.132
handling faceted objects 3.130
how to use 3.128
Undo command 3.131
FACETING
on curved surfaces 3.51
File formats supported 2.26
FILE MENU
Close command (⌘ W) 1.82
Import command (⌘ I) 2.38
New command (⌘ N) 1.89
Open As command 1.91, 2.42
Open command (⌘ O) 1.90, 2.2, 2.41
Page Setup command 1.93
Print command (⌘ P) 1.104
Quit command (⌘ Q 1.105
Revert to Saved 2.6
Revert to Saved command 1.106
Save A Copy As command 1.106
Save As 1.107
Save command (⌘ S) 1.107, 2.5, 2.6
FILTERS
anti-aliasing 8.75
third-party 8.63
using with animations 8.64
Fit Views to All command 1.88
Fit Views to Selection command 1.88
Flat Shading renderer 8.47, 8.73
FORMATS
file formats supported 2.26
importing and exporting 2.24
Frame differencing 7.63
Full-screen Playback command 7.79

G

GELS
built-in 6.19
custom 6.20
directional light sources 5.85
making your own 5.83
spot and point light sources 5.86
using pre-built gels 5.83
using regular textures 5.84
General Operation (introduction) 1.1
Gouraud renderer 8.47, 8.73

H

Handling textures in a model 5.50
Hidden Line renderer 8.74
Hierarchical scripting 7.10

I

IGES (file format) 2.27
Illustrator files 2.28
Image spooling settings 1.26
Import command (File menu, ⌘ I) 2.38
Import/Export (overview) 2.16, 2.24
IMPORTING
DXF files 2.26
EPS files 2.27
files 3.3
IGES files 2.27
Illustrator files 2.28
MiniCAD files 2.31
INSTANCES
saving memory 4.3
saving time 4.3
Internal image handling 8.10

K

Kerning text 3.11

L

LATHE
  angle of axis 3.143
  description 3.138
  how to use 3.140
  lathe axis 3.141
  multiple objects 3.140
  number of segments 3.143
  re-lathe option 3.145
  rotation 3.142
Launching Vision 3d 1.3
Lens perspective, default 1.21
Light meter 6.10
LIGHTING
  applying gels 6.19
  built-in gels 6.19
  custom gels 6.20
  directional 6.1, 6.4
  global 6.1, 6.4
  overflow handling 6.40
  palette 6.7
  point 6.1
  point light sources 6.11
  properties of 6.2
  setting attenuation 6.13, 6.18
  setting color 6.12, 6.16
  setting cone angle 6.16
  setting intensity 6.7, 6.13, 6.16
  spot 6.1
  spot light sources 6.15
Lighting control commands 6.39
Lighting Palette 1.16
Lighting palette 6.7, 6.8
LIGHTS
  setting color 6.6
Lights - default color 5.82
Linking tools 3.84
Loading existing models 2.2

M

Magnify/Zoom tool 3.62
Making curved surfaces from flat patches 3.56
Managing documents (introduction) 2.1
Managing lighting (introduction) 6.1
MAPS
  application methods 5.53
  Change Mapping dialog 5.55
  coverage vs. scale 5.55, 5.95
  determining the best resolution for the output 5.58
  example 5.56
  how big can a map be? 5.63
  how much memory will maps take? 5.64
  image resolution 5.57
  mapping variations 5.101
  re-sampling surface maps 5.62
  repetitions 5.100
  resolution in animations 5.62
  tile settings 5.99
Menu bar 1.5
MENUS (overview)
  Animation 1.11
  Apple 1.6
  Edit 1.8
  File 1.7
  Modeling 1.10
  Rendering 1.12
  Windows 1.12
MiniCAD (file format) 2.31
MiniCAD - bringing files into Vision 2.32
MiniCAD - exporting from older versions 2.31
Modelers included with Vision 3d 3.2
MODELING
  adjusting for scale differences 3.47
  built-in tools 3.1
  commands and controls used 3.161
  Extensions 3.93
  Fundamental elements of construction 3.49
  Merging models 3.47
  Methodology 3.45

MODELING MENU
Align command 3.162
Browse Shapes command 4.35
Browse Textures command 5.92
Edit Primitive command 6.39
Edit Primitive command (⌘ B) 3.164
Group command (⌘ G) 3.165
Lighting Effects command 6.39
Lock command 3.166
Modify command 3.169
Re-center command 3.179
Transform command 3.184
Ungroup command 3.185
Unlink command 3.185
Unlock command 3.186
Modeling objects 3.1
MODELING WINDOW
settings 1.21
MODELS
Closing 2.6
Creating new 2.1
Loading existing 2.2
Opening 2.1
Saving 2.4
Modifying bézier objects 3.116
Moving objects by dragging handles 3.68
Multiple view sets 1.47

N

New command (File menu, ⌘ N) 1.89
New Event Marker command (Animation menu) 7.75
New Window command (Windows menu, ⌘ /)1.90

O

Object Manipulation tools 3.64
Object Move tool 3.67
OBJECT ORIGIN POINT
description 3.170
moving 3.171
Object Rotate tool 3.70
Objects imported from other applications 3.3
Open command (File menu, ⌘ O) 2.41
OPENING
images and animations 2.14
models 2.1
Opening an image with an optimized palette 1.110
Opening and saving non-Vision files 2.23
Opening images and animations 2.14
Opening models 2.1
Orthographic view - use while modeling 1.32

P

Page Setup command (File menu) 1.93
PALETTES
Animation 7.37
Button bar 1.14
Lighting 1.16
Shapes palette 1.17, 4.30
Textures 1.16, 5.87
controls on the palette 5.90
Tools 1.15, 3.57
Palettes and buttons (general discussion) 1.13
Perspective control in camera windows 1.60
Perspective control in modeling windows 1.60
Phong Shading renderer 8.48, 8.74
Phong with Shadows renderer 8.50, 8.74
PICS (file format) 2.33
PICT (file format) 2.34
Play Sound command 7.78
Playback Controls sub-menu (Animation menu) 7.75
Playback Frame Rate command (Animation menu) 7.80
Point At 7.70
POINT LIGHTS
applying gels 5.86
illumination sources 6.11

modifier keys that apply 6.15
setting attenuation 6.13
setting color 6.12
setting intensity 6.13
POLYGONS
attaching normals 3.52
standard 3.53
triangulated mesh 3.53
PostScript printers - page setup 1.95
PREFERENCES
Enable Map Optimization 1.30, 1.103
saving defaults 1.98
setting 1.18
Previewing the work as you go 5.56
Primitives tools (3-D objects) 3.73
Print command (File menu, ⌘ P) 1.104

Q

QuickDraw printers - page setup 1.94
QUICKTIME
compression 1.109
explanation of compressors 7.65
movie (file format) 2.35
support 7.60

R

RayPainting Esoterica dialog 8.44
RayPainting renderer 8.50, 8.74
Raytracing Esoterica dialog 8.38
Raytracing renderer 8.49, 8.74
Re-animate command (Animation menu) 7.81
Relative and absolute references 1.35
Relative coordinates 1.35
RENDERERS
Flat Shading 8.47, 8.73
Gouraud Shading 8.47, 8.73
Hidden Line Removed 8.45, 8.74
Phong Shading 8.48, 8.74
Phong with Shadows 8.50, 8.74
Quick Shading 8.46, 8.73
RayPainting™ 8.41, 8.50, 8.74
Raytracing 8.36, 8.49, 8.74
RenderScript™ 8.51, 8.74
Wireframe 8.45, 8.73
RENDERING
Alpha channel support 8.62
animations 7.30
background 8.11
choosing the appropriate method 8.35
deleting preset rendering options 8.19
full-color 8.71
image quality 8.74
images (introduction) 8.1
internal image handling 8.10
perspective 8.8
pixel averaging method 8.25
playing back partially-completed animations 8.15
saving options by name 8.17
setting rendering options 8.16
settings 1.19, 3.91
spooling 1.28
super sampling method 8.24
suspending and restarting 2.9
suspending and restarting 8.12
windows 2.8, 8.27
RENDERING EFFECTS
base resolution 8.22
bump mapping 8.70
comparison chart 8.52
fog 8.71
frame size 8.21
glowing Surfaces 8.69
options 8.3, 8.19
quality level 8.22
reflectivity 8.69
refraction 8.70
setting parameters 8.20
shadows 8.69
smooth shading 8.70
surface mapping 8.68

Transparency 8.70
RENDERING MENU
Environment Effects command (⌘ H) 8.54
Render command (⌘ Y) 8.65
Render Using command 8.67
Rendering Effects command (⌘ K) 8.68
Rendering Options command 8.72
Rendering Queue command 8.75
Suspend Rendering command 8.77
RENDERING WINDOWS
controls 1.71, 1.72
RenderScript™ renderer 8.51, 8.75
Replicate command 3.21
Replicate command (Edit menu, ⌘ R) 3.179
Reverse Sequence command (Animation menu) 7.82
Revert to Saved command 2.6
RIB (file format) 2.37
Rotoscoper 5.50

S

SAVING
a copy of a model 2.6
images 2.7
models 2.4
PICS files (animation) 2.12, 2.13
QuickTime Movie 2.13
still images 2.10
Saving a copy of a model 2.6
Saving images 2.7
Saving models 2.4
Saving optimized palettes with images and animations 1.108
Scale tool 3.72
SCRIPTING
animations 7.1
example with simple actions 7.11
Select All command (Selection sub-menu, ⌘ A) 3.180
Selection sub-menu 1.9
SHAPES
adjusting instance scale during insertion 4.28
an example 4.4
benefits 4.8
creating new 4.26
deleting 4.30
handling shapes in a model 4.26
how they work 4.1
loading 4.26
making your own shapes for the Shapes palette 4.9
palette 4.30
root 4.7
saving 4.30
summary 4.8
use of (introduction) 4.1
Shapes palette 1.17
Show Animation Paths command (Animation menu) 7.83
Show/Hide Animation Palette commands (Windows menu) 7.82
Show/Hide palette buttons 3.92
Show/Hide Textures Palette command (Windows menu) 5.114
SKIN/UNSKIN
description 3.148
FlexObjects 3.149
tutorial 3.19
two-sided surface 3.148
using the modeler 3.151
using the Unskin modeler 3.151
SMOOTH
breaking angle 3.33, 3.135
description 3.134
flipping faces 3.140
how to use 3.135
limitations 3.138
smoothing curved surfaces with flat polygons 3.52

triangulated surfaces 3.54
when to use 3.133
Snapshot Windows (Windows menu) 3.183
SPOOLING
for renderings 1.103
renderings 1.28
sculpted objects 1.28, 1.103
surface maps 1.28, 1.103
SPOT LIGHTS
applying gels 5.86
attenuation 6.18
color 6.16
cone angle 6.16
intensity 6.16
modifier keys that apply 6.18
soft edges 6.17
sources 6.15
Strata Units 1.31
Super 3d (file format) 2.36
Surface Mapping 5.111
Surface maps — explanation of fields 5.21
SWEEP
axis 3.156
description 3.153
dialog box 3.155
how to use 3.155
multiple objects 3.154
number of steps 3.159
re-sweep option 3.160
step transformations 3.157
Tutorial 3.26
Switching tools from the keyboard 3.65
Swivel 3d (file format) 2.36

T

Text 3.78
Texture extensions (explained) 5.46
Texture transformation 5.102
TEXTURES
advantages of surface textures 5.3
application methods 5.54
applying 5.54
bump maps 5.25
color or grayscale maps 5.23
composite maps 5.32
controls on the palette 5.90
creating new textures 5.51
diffuse maps 5.26
Expert level dialog 5.10
Surface Properties 5.10, 5.107
extensions explained 5.47
extensions required for solid textures 5.2
glow maps 5.27
Introduction 5.1
loading 5.51
loading and removing maps 5.44
making and editing 5.3
moving on objects 5.102
reflectivity maps 5.27
registering multiple maps 5.30
rotating maps on objects 5.103
Solid or surface textures 5.2
solid textures 5.45, 5.49
specular maps 5.27
Surface properties
ambient fraction 5.11
bump amplitude 5.12
diffuse fraction 5.12
glow factor 5.13
index of refraction 5.14
reflectivity 5.14
smoothness/weight percentage 5.16
specular fraction 5.16
transparency 5.16
transparency maps 5.27
wireframe images as templates 5.33
TEXTURES EXAMPLE 5.34
ambient and diffuse maps 5.38
applying the texture 5.42
color map 5.37
expert level 5.41
making the maps 5.36
planning the mapping 5.35

putting it together 5.39
rendering the bottle 5.43
setting map size 5.35
setting surface properties 5.41
setting the mapping parameters 5.42
transparency map 5.37
Textures on light sources 5.81
Textures Palette 5.87
Textures palette 1.16
Textures palette (description and operation) 5.87
TIFF (file format) 2.37
Tool palette 1.15, 3.57
TOOLS
3-D Cone 3.74
3-D Cube 3.75
3-D Cylinder 3.74
3-D Sphere 3.74
Camera 3.88
Closed polygon 3.78
Irregular object 3.78
Light Source 3.85
Linking 3.83
Magnify/Zoom 3.62
Object Move 3.67
Object Rotate 3.70
Open polyline 3.78
Oval/circle 3.77
Primitives 3.73
Regular-sided Polygon 3.77
Round-corner rectangle/square 3.78
Scale 3.72
Square-corner Rectangle or Square 3.76
Text 3.78
Unlink 3.84
View Move/Grabber/Scrolling 3.59
View Rotate 3.60
Transform dialog box 1.36
Transformations and bounding boxes 3.46
TUTORIALS
for Chapter 1 1.38
for Chapter 2 2.17
for Chapter 3 3.3
for Chapter 4 4.10
for Chapter 5 5.64
for Chapter 6 6.21
for Chapter 7 7.40
for Chapter 8 8.29
Typical project sequence 1.1

V

Vertex points 3.50
View Magnify tool - modifier keys 3.64
View Management controls 1.46
View Management tools 3.58
View Move tool - modifier keys 3.60
View Move/Grabber/Scrolling tool 3.59
View Rotate tool 3.60
View Rotate tool - modifier keys 3.61
View set 1.46
View Set center 1.47, 1.48
VIEWS
controls 1.69
magnifying objects 1.63
managing a view set 1.48
modifier keys that apply to moving 1.49
modifier keys that apply to rotating 1.53
moving a view set center 1.49
multiple view sets 1.47
opening and closing 1.68
perspective control 1.60
rotating around the model 1.49

W

Window arrangement, default 1.23
Windows and views 1.38
WINDOWS MENU
Actual Size command 3.161
Camera Windows 3.163
Fit Views to All command (⌘ =) 1.88
Fit Views to Selection command (⌘ -)1.88
New Camera 3.171
New Window command (⌘ /) 1.90

Open Window command 1.92
Palettes command 1.112
Save Window command 1.107
Show/Hide Animation Palette commands 7.82
Show/Hide Lighting Palette command 6.41
Snapshot Windows 3.183
Split View command 1.113
Wireframe renderer 8.45, 8.73
Working with multiple models 2.4